

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ___ ” _____ 2021 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти

на тему

**“Дослідження та програмна реалізація системи VoIP для моделі
SaaS”**

Виконав здобувач вищої освіти

II курсу, групи КІ-20М-1,4

ОПП «Комп’ютерна інженерія»

спеціальності 123 «Комп’ютерна інженерія»

Якимчук В.В.

« ___ » _____ 2021 р.

Керівник проекту

кандидат технічних наук

Костянтин БУРАВЧЕНКО

« ___ » _____ 2021 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань . 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.

Олексій СМІРНОВ
« 6 » вересня 2021 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Якимчуку В'ячеславу Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи VoIP для моделі SaaS*

2. Керівник роботи *Буравченко Костянтин Олегович, канд. техн. наук*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 42-13 від 02.08.2021 року

3. Строк подання студентом роботи до захисту *10.12.2021 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи VoIP для моделі SaaS*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

7. Економічна ефективність розробленої

2. Перегляд аналогічних існуючих систем.

програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Наукова новизна

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2021	14.11.2021
Охорона праці	Оришака О.В.	06.10.2021	16.11.2021

7. Дата видачі завдання « 6 » вересня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2021 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2021 р.	
3.	Розробка моделі компонента	20.10.2021 р.	
4.	Розробка структур даних	25.10.2021 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2021 р.	
6.	Програмування алгоритмів	10.11.2021 р.	
7.	Розрахунок економічної ефективності	13.11.2021 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2021 р.	
9.	Оформлення ПЗ	17.11.2021 р.	
10.	Попередній захист роботи	10.12.2021 р.	

Дата видачі завдання
« 6 » вересня 2021 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2021 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Якимчук В.В. Дослідження та програмна реалізація системи VoIP для моделі SaaS. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2021.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи VoIP для моделі SaaS.

Метою розробки є дослідження та програмна реалізація системи VoIP для моделі SaaS.

Об'єктом дослідження є процес VoIP для моделі SaaS.

Предметом дослідження є методи VoIP для моделі SaaS.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи VoIP для моделі SaaS.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows XP/Vista/7/8/10.

Програму розроблено в середовищі Embarcadero RAD Studio Delphi 10.3.2 Rio Architect.

Ключові слова: комп'ютерна інженерія, VoIP, SaaS

ABSTRACT

Yakymchuk VV Research and software implementation of VoIP system for SaaS model. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2021

In this final qualifying work for the second (master's) level of higher education, software has been developed that is designed for the VoIP system for the SaaS model.

The purpose of the development is to research and software implementation of the VoIP system for the SaaS model.

The object of research is the VoIP process for the SaaS model.

The subject of research is VoIP methods for the SaaS model.

Research methods are based on methods of information theory and coding, methods of mathematical statistics, methods of software development.

The result is a software implementation of the VoIP system for the SaaS model.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

Developed user-friendly interface. Instructions for working with software are given.

The program can be used on a PC IBM PC with Windows XP / Vista / 7/8/10.

The program is developed in the environment Embarcadero RAD Studio Delphi 10.3.2 Rio Architect.

Keywords: computer engineering, VoIP, SaaS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	5
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ.....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	44
2.3 Розгорнута постановка завдання	48
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	50
3.1 Опис функціонування системи.....	50
3.2 Розробка структурної схеми	59
3.3 Розробка функціональної схеми.....	65
3.4 Розробка діаграми процесів	71
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ	74
4.1 Розробка блок-схем та опис алгоритмів функціонування системи	74
4.2 Захист розробленого програмного забезпечення	86
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ.....	89
6 НАУКОВА НОВИЗНА	95

ВКРМ-123.21.0029.00.00.ПЗ

Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Якимчук В.В.			Дослідження та програмна реалізація системи VoIP для моделі SaaS	Лім.	Аркуш	Аркушів
Перев.		Буравченко К.О.				М	1	132
Н.контр.		Гермак В.С.			ЦНТУ КІ-20М-1,4			
Затв.		Смірнов О.А.						

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	96
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.	96
7.2 Розрахунок трудомісткості розробки програмної продукції	98
7.3 Визначення чисельності виконавців і планового фонду зарплати	100
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника	105
7.5 Визначення собівартості розробки та ціни програмної продукції.	109
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	112
7.7 Визначення експлуатаційних витрат.....	113
7.8 Визначення економічної ефективності програмної продукції.....	114
7.9 Висновок.	116
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	117
8.1 Вступ.....	117
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером	118
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста	119
8.4 Розробка заходів з умов поліпшення охорони праці.....	122
8.5 Розрахункова частина	122
8.6 Висновки до розділу.....	124
9 ОСНОВНІ ВИСНОВКИ.....	125
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	127

MG	– Media Gateway. Транспортний шлюз
MGCP	– Media Gateway Control Protocol. Протокол керування шлюзами
MP	Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях
OSI	– Open System Interconnection. Взаємодія відкритих систем
PPP	– Point-to-Point Protocol. Протокол двостороннього зв'язку
RADIUS	Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданих їм послуг
RAS	Registration Admission and Status. Протокол взаємодії термінального устаткування з gatekeeper. Входить у сімейство протоколів H.323
RSVP	– Resource Reservation Protocol. Протокол резервування ресурсів
RTCP	Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі
SIP	– Session Initiation Protocol. Протокол ініціювання сеансів зв'язку
TAPI	Telephony Applications Programming Interface. Інтерфейс для програмування телефонних застосунків
TCP	Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP.
TCP/IP	– Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет
UDP	– User Datagram Protocol. Протокол передачі дейтаграмм користувача.
VoIP	– Voice over Internet Protocol. Технологія, що дозволяє використовувати IP-мережу для передачі мовної інформації

ВСТУП

Актуальність теми. VoIP (Voice over Internet Protocol) або IP -телефонія – це голосовий зв'язок через інтернет (у відмінності від традиційного телефонного зв'язку, що відбувається через телефонні лінії або мобільну GSM/3G/4G/5G мережу).

На даний момент основним призначенням IP-телефонії є дешеві або безкоштовні міжміські й міжнародні дзвінки. Для здійснення цих дзвінків вам потрібно скористатися послугами одного із провайдерів IP-телефонії й ви зможете дзвонити з комп'ютера, IP-телефону або звичайного телефону.

Однак, основна вигода VoIP для бізнесу – це можливість побудови більше ефективних систем корпоративних комунікацій з різними голосовими сервісами. Ефективність таких систем (у порівнянні із традиційними) полягає в наступному:

- більш просте й дешеве впровадження (так як VoIP системи будуються на базі існуючої інтранет-мережі);
- безкоштовний голосовий зв'язок усередині компанії (навіть при географічно розподіленій структурі бізнесу);
- можливість доступу до всіх комунікаційних можливостей будинку й у відрядженні (через інтернет);
- можливість інтеграції голосових сервісів у бізнес-застосунки й бізнес-процеси;
- просунуті можливості по записі розмов і веденню статистики.

Для реалізації систем корпоративних VoIP комунікацій служать IP-ATM і системи уніфікованих комунікацій на основі SaaS.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи VoIP для моделі SaaS.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

VoIP телефон також відомий як SIP телефон або софтфон, використовує передачу голосу по IP (VoIP) для здійснення телефонних дзвінків через мережу IP, а саме через інтернет. VoIP перетворює стандартний аудіоформат телефону в цифровий код, що може передаватися через інтернет, а також у зворотному порядку перетворює вхідні цифрові сигнали, що надходять із інтернет, у стандартні аудіосигнали телефону.

VoIP телефон дозволяє користувачеві дзвонити, використовуючи VoIP, на будь-який інший софтфон, мобільний або звичайний телефон. VoIP телефон може являти собою простий програмний софтфон, або пристрій, дуже схоже на звичайний телефон.

Функції VoIP телефону

От деякі зі стандартних функцій VoIP телефону: визначення номера абонента, що дзвонить, паркування, переведення і утримання виклику, доступ до телефонної книги й т.п. Деякі VoIP телефони також можуть під час розмови передавати відеозображення.

SIP телефон Windows/Mac – керування викликами з комп'ютера

SIP телефон Windows/MAC (софтфон) дозволяє через Інтернет управляти викликами з будь-якого місця у світі з настільного ПК або ноутбука. Незалежно від того, перебуваєте ви в офісі, удома або в дорозі, – софтфон для Windows або Mac значно скоротить витрати на зв'язок і працю системного адміністратора.

Завдяки відсутності витрат на настільні IP-телефони, гнучкості й потужним можливостям VoIP-технологій, софтфони для Mac або Windows будуть ідеальним рішенням для ваших бізнес-комунікацій.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

SIP телефон Windows/Mac для викликів із ПК

Користувачі VoIP-софтфонів для Windows і Mac можуть управляти викликами з робочого стола ПК перебуваючи як в офісі, так і віддалено. А з мультимедійною гарнітурою софтфон може повністю замінити настільний апарат.

- Виклики з робочого стола Windows, як на звичайному IP-телефоні.
- Доступ до всіх функцій уніфікованих комунікацій.
- Інтуїтивний інтерфейс – проста робота й керування.
- Ефективна робота поза офісом і економія на викликах.

Керування викликами в Панелі оператора

Софтфон включає функціональну Панель оператора, що настроюється для зручного оперативного керування викликами. П'ять режимів відображення інформації про виклики підійдуть для різних співробітників в організації.

- Переведення викликів простим перетаскуванням.
- Інформаційна панель і Менеджер черг для колл-центрів.
- Зручна індикація присутності співробітників.
- Режим секретаря для зручного керування вхідними викликами.

Інтеграція з MS Exchange, Outlook і іншими CRM

Дзвоните прямо зі своєї CRM-системи. Вхідний номер зіставляється з номером клієнта в базі CRM і фіксується в історії взаємин. Журналювання вхідних і вихідних викликів і тривалості кожного виклику дозволяє генерувати докладні звіти про дії ваших співробітників.

- Використовуйте MS Exchange, Microsoft Outlook або телефонний довідник АТМ.
- Дзвоните прямо з вашої CRM-системи.
- Заміна Caller ID зрозумілим ім'ям клієнта.
- Докладні звіти по клієнтах і співробітникам без ручного уведення.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

1.2 Область застосування

Областю застосування є SaaS. Модель SaaS (Software as a Service, що в перекладі означає «програмне забезпечення як послуга») – це система продажу програмного продукту, при якому доступ користувачеві надається через інтернет. Тобто замість того, щоб купувати й установлювати програмне забезпечення в себе на комп'ютері локально, сервіс доступний через всесвітню павутину або, як говорять, із хмари.

Користувач SaaS системи, що одержує доступ до застосунку, як би орендує його, оплачуючи певну суму за період часу. Завдяки цьому таке рішення є економічно вигідним. Але головна його перевага в тім, що в користувача немає необхідності займатися технічною стороною питання: установкою, підтримкою, відновленням, сумісністю й іншими питаннями – тобто йому залишається лише використовувати необхідного функціонала у своїх бізнес цілях.

Особливості SaaS системи

Перелічимо основні параметри SaaS, які відрізняють його від стандартного програмного забезпечення:

– Немає необхідності придбання ліцензії на використання продукту: замість цього оплачується його оренда на певний час. Це може бути щомісячна оплата або оплата за обсяг даних. При цьому сервісне обслуговування (підтримка й відновлення системи) уже включені у вартість.

– Одним сервісом може користуватися одночасно трохи клієнтів. Вони можуть мати до нього доступ з різних операційних систем і браузерів віддалено з будь-якої точки, де є інтернет-з'єднання.

– Якщо сервіс чимось не влаштував або необхідність його використання пропала, можна просто не продляти оплату послуги.

Сервісів на основі SaaS моделі в наш час існує досить багато, і всі вони мають свого клієнта. Однак разом з усіма своїми перевагами вони мають і певні недоліки. Саме про це ми й поговоримо нижче.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Переваги SaaS

Немає необхідності в установці програмного забезпечення на кожний комп'ютер – це основна перевага моделі, як уже говорилося вище.

– Скорочення фінансових витрат на придбання програмного продукту й наступну його підтримку.

– З боку розроблювача така модель дозволяє боротися із проблемою піратства – поширення неліцензійних копій програмного продукту, так як кінцева програма не попадає в руки користувача в готовому виді.

– Такі системи, як правило, є кроссплатформеними й кроссбраузерними, тобто не вимагають технічно певну операційну систему або браузер для роботи з додатком.

– Використання SaaS дозволяє не прив'язуватися співробітникам до робочого місця або комп'ютера: доступ до застосунку може бути здійснений з будь-якої точки з доступом до інтернету.

Недоліки SaaS

– Ваші комерційні дані при використанні SaaS системи будуть передаватися сторонньому провайдеру, що не завжди безпечно.

– Невисока швидкодія системи, що прямо залежить від швидкості інтернет-з'єднання.

– Через перебої з доступом до інтернету виникають простої в роботі, що досить ненадійно з погляду роботодавця.

Однак всі ці проблеми вже йдуть на другий план, так як сучасні технології дозволяють мати стабільний і швидкий доступ до інтернету, а технології шифрування даних дозволяють досить надійно здійснювати обмін комерційною інформацією за допомогою мережі інтернет. Саме тому SaaS системи одержують все більше поширення.

Користувачі SaaS

Основна частина користувачів SaaS – це малий і середній бізнес, якому досить накладно виходить придбання готового програмного продукту і його

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

наступна підтримка, тому більше вигідна його оренда. Крім того, така система буде вигідна для компаній із широкою мережею офісів або філій, між якими повинен здійснюватися постійний обмін даними, тому навіть технологія SaaS може зацікавити й великі компанії.

Приклади SaaS систем

Приведемо пару прикладів SaaS, якими ви вже могли користуватися або, напевно, знаєте.

- Корпоративна пошта на Gmail або інших клієнтах – це, мабуть, самий масовий і простий приклад SaaS технології.
- CRM і ERP – системи для керування проектами й ресурсами.
- Онлайн системи документообігу (той же google docs), органайзери, календарі – все це також приклади SaaS, хоча багато хто й безкоштовні.
- Послуги хостингу сайтів також яскравий приклад SaaS.

Сюди ж можна віднести й онлайн гри як сервіси, побудовані на тій же моделі, хоча їх і не прийнято класифікувати як SaaS. Що вже говорити про численні галузеві рішення, навіть у сфері розробки й розкручування сайтових величезна безліч: конструктори сайтів, системи автоматичного просування сайтів, біржі вічних і орендних посилань і багато чого іншого.

Альтернативи SaaS

Як ми говорили вище, SaaS не ідеальні, тому з'явилися деякі альтернативні рішення, засновані на її моделі, але які мають свою модифікацію:

- **Хмарні платформи.** Якщо ви не хочете віддавати свої комерційні дані сторонньому провайдеру, то вам буде вигідно орендувати не додаток, а комп'ютерні потужності, на які встановити придбане програмне забезпечення.
- **Хостинг застосунків.** Дана модель відрізняється від SaaS архітектурою серверної частини, тому для рядового користувача різниця буде не видна. Суть її в тім, що хостинг провайдер здійснює установку окремої копії застосунку для кожного клієнта замість обслуговування одночасно декількох користувачів. Цей

процес більш складно адмініструвати й виконувати відновлення ПЗ, тому й коштує така послуга дорожче.

– S+S – це модель від Microsoft, що пропонує для доступу до сервісу використовувати не браузер, а програмний клієнт.

Як ви могли зрозуміти, з технологією SaaS ви вже зіштовхувалися неодноразово, і кількість таких сервісів продовжує рости через їхню затребуваність і простоту використання. Але прогрес на цьому не зупиняється, пропонуючи усе більше зроблені й цікаві ідеї з урахуванням потреб кожного.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи VoIP для моделі SaaS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Технологія VoIP реалізує завдання й рішення, які за допомогою технології традиційної телефонії реалізувати сутужніше або дорожче. Метою написання даного розділу є бажання познайомити з рішеннями в області IP-телефонії від різних постачальників, які можуть бути використані в невеликих компаніях і філіях великих корпорацій

Ви володієте або керуєте невеликою компанією до 50 чіл? Компанія має мережу невеликих філій, розкиданих по країні або місту? Питання комунікації із зовнішнім миром (телефонні й мобільні дзвінки партнерам і замовникам, електронні листи, обмін миттєвими повідомленнями, Інтернет) є життєво важливою частиною бізнесу?

Ви зіштовхувалися із ситуацією, коли внаслідок зміни офісу або місцезнаходження філії компанії доводилося міняти телефонні номери, а виходить, втрачати клієнтів? У вас ніколи не виникала думка, що було б здорово, якби рахунку на зв'язок були хоча б вполовину менше? Якщо так, то ця робота – для вас.

IP-телефонія. За й проти

Розглянемо основні переваги впровадження IP-телефонії для невеликих компаній і те, яким образом вони досягаються:

1. Простота розгортання системи. Одним з основних переваг IP-телефонії є те, що голос передається по каналах передачі даних. Тобто при організації зв'язку в новому офісі немає необхідності готувати кабельну інфраструктуру окремо для локальної мережі й окремо для телефонів.

						ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			13

5. Економія на міжнародних дзвінках. Надається можливість скористатися послугами провайдерів IP-телефонії й робити міжнародні дзвінки по лояльних тарифах.

6. При підключенні через провайдерів існує можливість зафіксувати номерну ємність через SIP і у випадку зміни місця розташування офісу телефонні номери залишаються незмінними. У нашій країні вже достатня кількість операторів і провайдерів, які дозволяють підключати міські телефонні номери не по традиційних каналах (аналоговим лініям або за протоколам PRI), а по каналах IP (інакше кажучи, через Інтернет). Таким чином, навіть у випадку зміни офісу у вас завжди залишиться можливість зберегти свої номери телефонів. Єдина умова – у новому офісі теж повинен бути Інтернет.

Говорячи про переваги технології, несправедливо було б не сказати про так званих «хронічні хвороби» VoIP, про які відомо вже багато років, але усунути їх повністю неможливо. Можна лише звести до мінімуму ефект від їхнього впливу.

Порівняно повільне впровадження VoIP зв'язане не тільки з високими початковими інвестиціями. Мова йде також про наступні проблеми:

1. Гадана простота впровадження (багато в чому завдяки публікаціям у пресі й рекламі виробників). Необхідно не забувати, що розгортання голосових служб на базі IP-мережі підприємства все-таки вимагає участі фахівців.

2. Недостатня часом якість голосового зв'язку. Основну роль мають два фактори: характеристики IP-мережі і якість використовуваного устаткування. При використанні IP-мережі на якість передачі голосу впливають такі параметри, як затримка при передачі IP-пакетів, джиттер і втрата пакетів.

Ці фактори потрібно враховувати, і, як уже було сказано вище, їх можна звести до мінімуму (наприклад, як правило, необхідне проведення попереднього організаційно-технічного аналізу існуючої мережної інфраструктури).

Що ж стосується якості устаткування, те тут багато чого залежить від рівня виробника й досвіду системного інтегратора, що поставляє рішення. Хто ж

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

є сьогодні основним споживачем IP-телефонії? Звичайно, у першу чергу це самі оператори зв'язку. Далі йде великий бізнес: банки з їх численною філіальною мережею, корпорації й територіально розподілені компанії, мережі магазинів і т.д. У меншому ступені IP проникнула на ринок невеликих компаній. Там вона в основному представлена Skype. Хоча перспективи саме в даному сегменті, на думку аналітиків, у неї більші.

Специфіка роботи в сегменті SMB

Хто з постачальників устаткування й рішень хоч раз зіштовхувався з малим і середнім бізнесом, погодиться – це досить консервативний сегмент.

Ніде правди діти – хоча частина SMB-компаній ставить на перше місце надійність рішення або послуги, а на друге – параметр «ціна/якість», більшість скоріше дотримується зворотної моделі. І криза отут ні при чому. Просто, як правило, SMB-компанії не відносять покупку телекомунікаційних послуг до числа пріоритетних завдань.

У компаніях SMB у переважній більшості випадків відсутній окремий IT-бюджет, не говорячи вже про його фінансування на регулярній основі. У результаті й інформаційній системі, і інфраструктура складаються поступово, без єдиного плану. Та й сама стратегія розвитку (у випадку, якщо вона все-таки є) будується, як правило, на експертній оцінці прихожого фахівця, що обслуговує IT-інфраструктуру компанії. Не будемо проводити експертну оцінку рівня підготовки даних «експертів». Будемо вважати, що він досить високий, щоб оцінити переваги й недоліки того або іншого рішення й дати необхідні рекомендації своєму роботодавцеві.

Крім того, у компаніях SMB на ухвалення рішення сильний вплив робить суб'єктивний фактор: думка їхнього власника або керівника – «подобається, не подобається», «довіряю, не довіряю». Ні для кого не секрет, що не всі підприємства цього сектора прозорі, тому їхні перші особи не завжди готові пустити «чужого» у свою мережу.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Ключовою фігурою в цьому випадку є «перша особа» компанії і його готовність купувати нове обладнання, а отже, вкладати гроші в розвиток бізнесу.

Тому надто важливо пропонувати просте рішення, зрозуміле як з погляду необхідності того або іншого устаткування й послуг, так і з погляду оплати, підключення й експлуатації. Це рішення повинне бути зрозуміло людям, що найчастіше не знають, що таке, наприклад, DNS і IP-адреса.

Клієнти із сегмента SMB, як правило, більш оперативно приймають те або інше рішення, чим великі, однак, швидше за все, це все-таки не той сегмент, з якого треба починати впровадження технологічних новинок. До того ж в SMB-секторі, як правило, перевага віддаються типовим «коробковим» рішенням.

Однозначно можна сказати, що бездротові технології є найкращим рішенням для цього сегмента. Швидке підключення послуг, схоронність послуг при переїзді, безперечно, дуже привабливі для невеликих компаній.

Навіть при наявності досить великого переліку вимог, висунутих ринком, малий і середній бізнес залишається одним з найпривабливіших секторів розвитку для більшості виробників і постачальників телекомунікаційних рішень. Багато компаній прагнуть зробити свої продукти оптимальними й привабливими саме для SMB.

Dynamix

Для організації телефонного зв'язку із застосуванням технології VoIP для компаній і офісів на 30-40 користувачів компанія « Вектор-київ» пропонує IP-телефонні станції (IP PBX) Dynamix ePBX-100A-128 і Dynamix IP PBX-6200S.

ePBX-100A-128 – економічне рішення для компаній до 100 користувачів, надає повний спектр базових телефонних послуг, функції автосекретаря (auto attendant), організацію голосового меню (IVR), підтримку SIP-транків, інтеграцію з відеопристроями (відеотелефонами Dynamix DW VP-800), програмними телефонами.

Голосові повідомлення й інформація про дзвінки (CDR) зберігаються на убудованому CF-накопичувачі ємністю 1 Гб. Повна відсутність у конструкції

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

сервер не відповідає або буде отримана відповідь, що абонент не знайдений або недоступний (не зареєстрований на цьому сервері), шлюз автоматично повторить запит на другому. Таким чином, для забезпечення резервування можна встановити дві телефонні станції Dynamix ePBX-100A-128 з ідентичними налаштуваннями й всі пристрої настроїти на одночасну роботу із двома станціями.

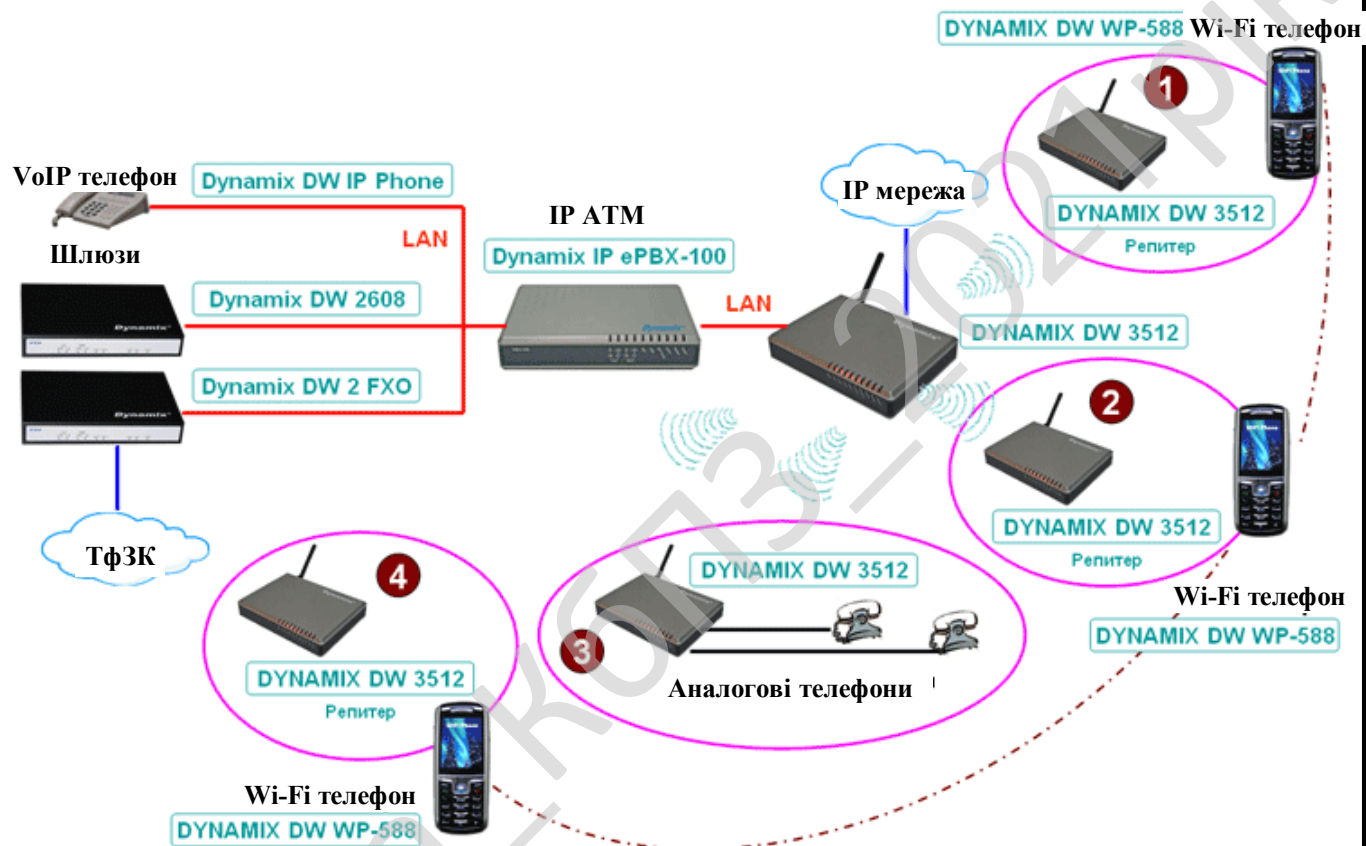


Рисунок 2.1 – IP-телефонні станції (IP PBX) Dynamix

Dynamix IP ePBX-100A-128 – IP АТМ з розширеними функціями

Dynamix IP ePBX-100A-128 – IP АТМ нового покоління для невеликих підприємств або офісів. Dynamix IP ePBX-100A-128 призначена для виконання ряду VoIP застосунків, наприклад, голосових і відео конференцій, передачі викликів, обслуговування вхідних викликів і IP-базованого зв'язку. Пристрій невеликого розміру дозволить підприємствам або офісам, що мають Internet доступ, здійснювати VoIP телефонію.

Користувачі також можуть вибрати й інше устаткування з VoIP сімейства Dynamix для одержання додаткових функцій при роботі з IP ATM Dynamix IP ePBX-100A-128. Для забезпечення функцій доступу до традиційних PSTN можна використовувати Dynamix DW 2 FXO або Dynamix DW 4 FXO. IP телефони Dynamix DW IP Phone і Dynamix DW Phone LP – 201 можуть забезпечити додаткове розширення функцій IP ATM. Гнучк і багатофункціональна IP ATM Dynamix IP ePBX-100A-128 дозволяє здійснити повний перехід від традиційних ATM до ATM нового покоління – IP-ATM.

Основні відмінності Dynamix IP ePBX-100A-128 від Dynamix IP ePBX-100A – це інтеграція з відеопристроями, наприклад, відеотелефонами Dynamix DW VP-800, а також підтримка SIP базованих відеокамер і систем відеоспостереження, що значно розширює функціональність даної ATM.

Особливості:

- Сумісність із SIP RFC3261.
- Для забезпечення функцій доступу до традиційних PSTN можна використовувати Dynamix DW 2 FXO або Dynamix DW 4 FXO.
- IP телефони Dynamix DW IP Phone і Dynamix DW Phone LP – 201 можуть забезпечити додаткове розширення функцій IP ATM.
- Підтримка функцій традиційних ATM.
- Підтримка фіксованого IP, DHCP мода й PPPo мода.
- Керування за допомогою Web Browser.
- Мережний інтерфейс: 1WAN і 1LAN.
- Підтримка голосових кодеків: G.729, G.711 μ , G.711A.
- Підтримка H.263 і MPEG4 відеокодеків (pass-through).
- до 100 користувачів, 15 конкуруючих дзвінків.
- Пам'ять 128 МБ RAM.
- 1 ГБ для голосових повідомлень.
- Asterisk V1.4.
- Підтримка T.38 FAX.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Протоколи

– SIP (Session Initiation Protocol).

Функції

– Автентифікація (Authentication).

– Передача виклику (Call Transfer) – {IP телефони, Dynamix DW FXS – 04}.

– "Сліпа" передача (Blind Transfer) – {IP телефони, Dynamix DW FXS – 04}.

– Переадресація виклику, якщо зайнято (Call Forward on Busy) – {IP телефони, Dynamix DW FXS – 04}.

– Переадресація виклику, якщо не відповідають (Call Forward on No Answer) – {IP телефони, Dynamix DW FXS – 04}.

– Безумовна переадресація (Call Forward Unconditional) – {IP телефони, Dynamix DW FXS – 04}.

– Утримання дзвінка (Call Hold/Retrieval) – {IP телефони, Dynamix DW FXS – 04}.

– Маршрутизація виклику (Call Routing).

– Очікування дзвінка (Call Waiting) {IP телефони, Dynamix DW FXS – 04}.

– Визначення Caller ID {IP телефони, Dynamix DW FXS – 04}.

– Не турбувати (DoS Not Disturb) {IP телефони}.

– Гнучка логіка розширення – Music On Hold.

– Музика під час переадресації.

– Паркування дзвінка.

– Трестороння конференція {IP телефони}.

– Час і дата.

– Trunking {Dynamix DW 4 FXO}.

– VoIP шлюзи {Dynamix DW 4 FXO}.

– Голос у пошту (Voice Mail to e-mail).

– Запис деталей дзвінків (Call Detail Records).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

Технічні особливості

- NAT трансляція.
- Привітальне повідомлення для телефонів.
- Керування: за допомогою Web Browser.
- HTTP upgrade firmware і файлів дзвінків.
- Експорт – імпорт конфігурації.
- Мережний інтерфейс: 1WAN і 1LAN.
- Підтримка голосових кодеків: G.729, G.711μ, G.711A.
- DTMF: у смузі, RFC 2833, SIP-Info.
- Мережа: Підтримка фіксованого IP, DHCP мода й PPPo мода.

Ємність

- 100 користувачів.
- 15 конкуруючих дзвінків.

Кодеки

- G.711 (A-Law & M-Law).
- G.729.
- H.263 і MPEG4 відеокодеків (pass-through).

Світлодіодна індикація

- **Power:** Світиться при включенні живлення.
- **Status:** Світиться, якщо система готова.
- **Alarm:** Мигає, якщо є проблеми.
- **SIP Trunk:** Світиться коли Dynamix IP PBX -100A-128 успішно

zareestruvav vsi mozhlivi SIP trunki;

Мигає, коли Dynamix IP PBX -100A-128 має помилки при реєстрації одного з SIP транків;

Не світиться, коли немає доступних SIP транків.

– **CDR:** Dynamix IP PBX-100A-128 може записати деталі дзвінка на зовнішньому комп'ютері. Користувач повинен виконати

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

CDR програму на комп'ютері, коли дані готові й Dynamix IP PBX -100A-128 готовий підключитися до CDR сервера.

Це стан індикується світлодіодом, який світиться.

– **NET:** Відображає мережний статус. Якщо WAN порт Dynamix IP PBX -100 працює в з фіксованої IP модою, світлодіод світиться. Якщо WAN порт працює в DHCP моді й Dynamix IP PBX -100 одержує IP, світлодіод мерехтить, а якщо WAN порт працює в DHCP моді й Dynamix IP PBX -100 не одержує IP – світлодіод гасне.

– **WAN**

LINK/ACT: Світиться коли WAN порт підключений до мережі. Мигає – при передачі даних.

10/100: Світиться, якщо швидкість у мережі 100 Mb/s, і гасне при швидкості 10 Mb/s.

– **LAN**

LINK/ACT: Світиться коли WAN порт підключений до мережі. Мигає – при передачі даних.

10/100: Світиться, якщо швидкість у мережі 100 Mb/s, і гасне при швидкості 10 Mb/s.

Фізичні параметри:

– Розміри: 17.5 x 12.5 x 3.2 см.

Застосування:

– SOHO телефонія.

– IP-PBX, як альтернатива звичайної АТМ.

– Передача IP відео.

– Телефонні дзвінки через IP мережу.

– Передача факсів через IP мережу.

Dynamix SIP PBX 6200x – IP АТМ для корпоративного застосування

Dynamix SIP PBX 6200x – IP АТМ нового покоління для великих підприємств, корпорацій або офісів. Dynamix SIP PBX 6200x призначена для

						ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			23

виконання ряду VoIP застосунків, наприклад, голосових конференцій, передачі викликів, обслуговування вхідних викликів і IP- базованого зв'язку. Ці АТМ можуть інтегруватися з відеодомофонами на вході в офіс, а так само з IP відеотелефонами, наприклад, Dynamix DW VP-800 і SIP IP відеокамерами, що дозволяє організувати не тільки відеозв'язок між співробітниками, але й забезпечити віддалене відеоспостереження в офісі або на підприємстві з використанням Web броузера.

Dynamix SIP PBX 6200S підтримують до 200 користувачів, Dynamix SIP PBX 6200GS – до 400, а Dynamix SIP PBX 6200N – до 1000, що може служити економічно вигідним рішенням переходу від традиційних АТМ до АТМ нового покоління – IP-АТМ для середніх і більших організацій.

Для одержання додаткових функцій при роботі з IP АТМ Dynamix SIP PBX 6200x може бути використане також і інше устаткування сімейства Dynamix. Для забезпечення функцій доступу до традиційних PSTN можна використовувати Dynamix DW 2 FXO або Dynamix DW 4 FXO. IP телефони Dynamix DW IP Phone й Dynamix DW Phone LP – 201 можуть забезпечити додаткове розширення функцій IP АТМ.

Особливості:

- Підтримка передачі даних, голосу й відео по однієї IP мережі.
- Підтримка до 1000 абонентів (з використанням різних моделей Dynamix SIP PBX).
- Сумісність із SIP RFC3261, а так само підтримка SIP/RTP шифрування/дешифрування для збільшення безпеки передачі інформації.
- Підтримка функцій традиційних АТМ, можливість здійснювати як VoIP, так і звичайні (PSTN) дзвінки.
- Інтеграція з відеодомофонами на вході в офіс, а так само з IP відеотелефонами, наприклад, Dynamix DW VP-800 і SIP IP відеокамерами.
- Керування за допомогою Web Browser.
- Підтримка голосових кодеків: G.729, G.711 і GSM.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

- Підтримка RADIUS Server або Enterprise Billing через TCP.
- Підтримка High Available active/standby Redundant (опціонально).
- Підтримка SIP ENUM.
- Підтримка VLAN і QOS tagging (Dynamix 6200N не підтримує VLAN).
- Підтримка Setup Wizard для простої інсталяції з IP телефонами.
- Убудована функція CDR.

Протоколи

- **SIP (Session Initiation Protocol) RFC 3261**

Функції автосупроводження (Auto-attendant)

- **Web-базований редактор потоків автосупроводження (Web-Base Auto Attendant Flow Editor).**
- Підтримка спеціальних оголошень за розкладом (Scheduled Special Announcement).
- Підтримка святкових і робочих днів (Holidays Working Time Support).
- Підтримка декількох мов (Multiple Language Support).
- Підтримка філій (Support Branch Office).
- Підтримка транзитних викликів (Support Transit Call).

Голосова пошта (Voice Mail)

- **Web-базований редактор голосових потоків (Web-Base Voice Mail Flow Editor).**
- Персональне вітання (Personal Greeting).
- Підтримка декількох мов (Multiple Language Support).
- Підтримка TTS – перетворення тексту в мову (Native TTS [Chinese & English & Japanese] Support).
- Підтримка TTS на додаткових мовах (Support Additional Customized TTS Language).
- Індикація повідомлень, що очікують (Message Waiting Indication).
- Повідомлення про пошту (Email Notify).
- Web пошук (Web Retrieve).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

– Пошук телефонів (Phone Retrieve).

Міст конференцій (Conference Bridge)

– Підтримка RFC 4579 (без XML).

– Спеціальні конференції (Ad-Hoc Conference).

– Віртуальні конференції (Virtual Conference).

– Спеціальні віртуальні конференції (Virtual Conference – Ad-hoc).

– Тон повідомлення про події (Event Tone Notice).

– до 8 учасників для ПЗ DSP (Up to 8 parties for software DSP version)

[Dynamix 6200S і Dynamix 6200GS].

– Швидкі конференції для Soft-телефонів (Quick Conference by Soft-phone).

Розширені сервіси (Enhanced Service)

– Сервіси анонсів (System Announcement Service).

– Різні тони для різних компаній (Company-wide Coloring Ring Back Tone Service).

– Надає сервер тонів утримання (Provided Server Hold Tone).

Маршрутизатор голосу (Voice Router)

– Державні й частки (Public and Private IP Legs).

– SIP-Aware RTP Routing.

– VOIP Firewall/NAT.

Опціональні функції

– Переносний Soft-phone.

– Оператор консолі ПЗ (Operator Console Software).

– Система білінгу для підприємств (Enterprise Billing Software).

– Модуль Web-дзвінків (Web Caller Module).

– Модуль Microsoft Office Communicator/Exchange 2007 Module (опція).

Телефонні функції

– Переведення виклику (Call Transfer).

– Автоматична переадресація дзвінка (Call Forward).

– Повідомлення про переадресацію дзвінка (Call Forwarded Notice).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Відображення дзвінка (Call Screening).
- Таємність ID що дзвонить (Caller ID Privacy).
- Очікування дзвінка (Call Waiting).
- Утримання дзвінка (Call Hold_).
- Відновлення дзвінка (Call Pickup – Global, Group).
- Спеціальне відновлення дзвінка (Specified Call Pickup).
- "Знайди мене" (Find Me).
- Короткий код (Short Code).
- Не турбувати (DoS Not Disturb).
- Повідомлення по Email про пропущений дзвінок (Miss Call Notify by Email).
- Заміна ANI – автоматичного ідентифікатора номера (ANI Replacement).
- Повернення дзвінка (Call Return).
- Сховати/показати ANI (Hide ANI/Show ANI Selection).
- Паркування/пошук дзвінка (Call Park/Retrieve).
- Показ заміненого номера (Display Name Replacement).
- Номер PSTN (Number).
- Одночасний дзвінок PSTN і IP (Ring PSTN & IP Device Simultaneously).
- Широкополосні сервіси (Broadcasting Service).
- Дзвінок будильника (Wake-Up Call).
- Видалення анонімних дзвінків (Reject Anonymous Call).
- Підтримка SIP TAPI (Support SIP TAPI).
- Busy Lamp Filed (RFC 4235).

Фізичні й параметри:

Dynamix PBX 6200N:

- Розміри: 19-дюймів, 2U.
- Вага: 15 кг.
- Один Gigabit WAN порт і Gigabit LAN порт.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Dynamix PBX 6200GS:

- Розміри: 19-дюймів, 1U.
- Вага: 12 кг.
- Один 10/100 Base-T WAN порт і 10/100 Base-T LAN порт.

Dynamix PBX 6200S:

- Розміри: 25 x 17 x 4,5 див.
- Вага: 2,5 кг.
- Один 10/100 Base-T WAN порт і 10/100 Base-T LAN порт.

Параметри навколишнього середовища

- Вологість 10-95% (без конденсату).
- Робоча температура від 0 °C до +50 °C.

Живлення:

- Вхідне: AC100V~240V, 50-60Hz.

VoIP шлюзи:

Dynamix ATA-171P – 1 FXS і 1 PSTN порт

Dynamix ATA-171/171P/172 /171M – VoIP шлюзи, призначені для передачі голосових даних високої якості й факсів по IP мережах. VoIP шлюзи для стиску мови використовують найпоширеніші кодеки: G.711, G.726, G.729A і G.729B, підтримує SIP протокол.

VoIP шлюзи мають один (Dynamix ATA -171, Dynamix ATA-171P, ATA-171M) або два FXS порти (Dynamix ATA-172), а модель Dynamix ATA-171P підтримує також PSTN backup (при провалі IP з'єднання переходить автоматично на PSTN лінію).

Dynamix ATA-171M, крім FXS порту має також один FXO порт і забезпечує передачу виклику між FXS, FXO і IP портами (Call Transfer) і пересилання між FXS, FXO і IP портами (Call Forwarding).

Нову серію VoIP шлюзів Dynamix ATA відрізняють малі габарити (99 X 99 X 32 мм).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Переваги

- Легкий доступ до IP через телефон або міні-АТМ.
- Зменшення вартості телефонних розмов.
- Підтримка існуючого плану телефонних номерів.
- Просте підключення через ADSL/Cable Modem і інше устаткування для роботи на виділених лініях.

- Сумісність із Asterisk V1.2, V1.4.

Особливості:

- Підтримує SIP протокол.
- Автоматичний вибір IP або PSTN (Dynamix АТА-171Р, Dynamix АТА-171М).
- Автоматичне перемикавання на телефонну лінію при ушкодженнях IP мережі (Dynamix АТА-171Р).

- Генерація аналогових телефонних сигналів (готовність, зайнято, утримання дзвінка, повернення дзвінка).

- PPPo.

- Підтримка NAT/DHCP.

- Визначення/генерація тонових сигналів DTMF.

- Відновлення ПЗ через TFTP, Console, HTTP.

- Віддалене конфігурування/скидання.

- Світлодіодні індикатори.

- Підтримка статичних IP-адрес і DHCP.

- Повна підтримка аналогових телефонних сигналів.

Мережні протоколи

- SIP v1 (RFC2543), v2 (RFC3261).

- IP/TCP/UDP/RTP/RTCP.

- IP/ICMP/ARP/RARP/SNTP.

- TFTP Client/DHCP Client/ PPPo Client.

- Telnet/HTTP Server.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- DNS Client.
- NAT/DHCP Server.

Голосові кодеки

- G.711:64kbit/s (PCM).
- G.726:16k/24k/32k/40kbit/s (ADPCM).
- G.729A: 8k bit/s (CS-ACELP).
- G.729B: VAD & CNG в G.729.

Особливості голосу

- VAD: Voice activity detection (Визначення пауз).
- CNG: Comfortable noise generator (Генерація шуму в паузах).
- LEC: Line echo canceller (Лунапридушення).
- Packet Loss Compensation (Компенсація загублених пакетів).
- Adaptive Jitter Buffer (Адаптивний буфер джиттера).
- Генерація тонів: Ring Tone , Ring Back Tone , Dial Tone, Busy Tone,

Programming Tone.

- Підтримка T.38 факс.

Дзвінки

- Call Hold (Утримання дзвінка).
- Call Waiting (Очікування дзвінка).
- Call Forward (Переадресація виклику).
- Caller ID (Визначення Caller ID).
- 3-way conference (3-х стороння конференція).

Призначення IP

- Static IP.
- DHCP.
- PPPo.

Безпека

- HTTP 1.1 базова автентифікація для Web setup.
- MD5 для SIP автентифікації (RFC2069/ RFC 2617).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Відновлення мікропрограм і конфігурація

- Web Browser.
- Telnet.
- TFTP.
- HTTP.

Інтерфейси

- Два RJ-45 10/100 Base-T Ethernet порти – одні для підключення до ADSL модему або комутатора (WAN), а другий для підключення комп'ютера (LAN).
- Один RJ-11 "Phone" (FXS) для підключення телефону (Dynamix ATA 171, Dynamix ATA-171M і Dynamix ATA-171P).
- Два RJ-11 "Phone1 і Phone2" (FXS) для підключення телефонів (Dynamix ATA-172).
- Один RJ-11 "PSTN " для підключення PSTN (тільки Dynamix ATA-171P).
- Один RJ-11 "Line" (FXO) – Dynamix ATA-171M.
- Живлення: DC 12V (зовнішній блок живлення, 100-240V AC).

Розмір

- 99 X 99 X 32 мм.

Параметри навколишнього середовища

- Робоча температура: від 0° C до 40° C.
- Вологість: від 10% до 90%.

Застосування:

- ISP/ITSP (Internet Telephony Service Provider).
- Корпоративна телефонія.
- SOHO телефонія.
- IP-PBX, як альтернатива звичайної АТМ.
- Телефонні дзвінки через IP мережі.
- Передача факсів через IP мережі.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

- Підтримка контролю доступу за допомогою ANI, DNIS, IP, тільки гейткипер, тільки проксі або тільки RADIUS.
- Розпізнавання SIP дзвінків через NAT.
- Підтримка протоколу передачі факсів T.38 fax relay до 14400 bps у режимах SIP і H.323.
- Убудований IVR і контроль потоку дзвінків.
- Сумісність із устаткуванням Cisco.
- Збереження конфігурацій.
- Надає CDR (Call Detail Recording).
- Убудований режим ідентифікації prepaid і postpaid користувачів.
- H.323 і SIP конверсія сигналів.
- Сумісність із Asterisk V1.2, V1.4.

Інтерфейс

- Два 10/100MB Ethernet порти (Host & VoIP потік).
- До чотирьох T1/E1 (збалансовані 120 Ohm, RJ48C).

Протоколи й стандарти

Сумісність:

- ITU-T H.323 v2/v3/v4.
- SIP RFC 2543/3261.
- H.450 (Call Hold, Call Transfer, Call forward).

Властивості аудіо

- Кодеки – G.711A/μ-Law, G.723.1 (5.3K/6.3K), G.729A.
- Режим лунапридушення G.168.
- Підтримка безшумного режиму придушення для G.729A, G.723.
- VAD (Voice Activity Detection) – визначення пауз.
- CNG (comfort Noise Generate) – генерація шуму в паузах.

DTMF передача

- Прозора (transparent).
- H.245 сигнальний/буквено-цифровий.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

– H.323 Q.931.

– RFC 2833.

– SIP INFO.

FAX підтримка

– Автоматичне виявлення факсу.

– T.38 fax relay, засноване на H.323 Annex D.

– SIP T.38 fax relay.

– До G3 FAX.

– ECM підтримка.

– Сумісність із устаткуванням Cisco.

Убудовані IVR і контроль потоку дзвінків

– Повний контроль дзвінків (одноетапний або багатоетапний набір).

– IVR функції.

Особливості керування

– Можливість відновлення OS і прошивання.

– Консольний порт: RS-232.

– TELNET.

– Повне Web керування й монітор реального часу.

– Керування з передньої панелі (PKI й кнопки).

– Підтримка SNMP v2 (H.341. і SNMP v4.

– Керування обліковим записом користувача.

– Визначення тимчасової зони.

– Підтримка фіксованих IP і DHCP.

– Підтримка DNS і Dynamic DNS.

Світлодіодні й PKI індикатори

– POWER: Індикатор живлення, звертання до запам'ятовувального пристрою.

– Передня панель PKI (2 рядка по 16 символів).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Живлення

– 90~240V авто перемикач.

Параметри навколишнього середовища

– Робоча температура: 0°C – 60°C.

– Відносна вологість: 5% – 95%.

Фізичні параметри

– Розміри: 483 мм(ш)*44мм(в)*450мм(д).

– Вага: 8 кг.

– 1U.

Dynamix DW 3512

VoIP маршрутизатор з 2 FXS портами, WAN і PSTN портами, 4 LAN і підтримкою WiFi 802.11 b/g

Dynamix DW 3512 – VoIP WiFi шлюз, що має 2 FXS порти, 1 PSTN порт, 1 WAN/4 LAN і 802.11 b/g WLAN інтерфейси. Цей новий VoIP шлюз підтримує SIPv2 протокол (RFC3261).

Підтримка PSTN інтерфейсу забезпечує автоматичний або ручний перехід на PSTN мережа, якщо з'являються проблеми з IP мережею. Підтримка бездротових мереж у режимі точки доступу або клієнта (Access Point, Client mode) забезпечує автоматичне сканування й керування доступом, засноване на MAC адресах. Підтримка WDS і універсального повторювача (Universal Repeater) забезпечує об'єднання декількох бездротових мереж.

Підтримка бездротових мереж – важлива особливість даних шлюзів, що особливо затребувана ISP провайдерами для створення комбінованих мереж з використанням бездротових технологій.

Особливості:

– Дизайн "усе в одному" дозволяє значно скоротити витрати на устаткування.

– Застосування для невеликих офісів і користувачів, що забезпечує одночасно підтримку WiFi AP, WiFi Client, IP і VoIP шлюзів.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

- Підтримка як WDS, так і універсального повторювача для простого об'єднання бездротових мереж.
- Бездротовий, простий ISP набір для підключення до різних бездротових мереж ISP провайдерів.
- Інтеграція з IP-АТМ (Dynamix PBX-100) і WiFi телефонами для створення бездротової офісної АТМ системи WiFi – IP-PBX.
- Підтримка зовнішніх зв'язків через ADSL, кабельний модем або виділену лінію.
- Підтримка протоколу Multiple VPN pass-through для підключення до існуючої VPN мереж.
- Гнучкий план абонентів для VoIP, PSTN і бездротових викликів.
- Безпека, підтримка Firewall і захист від відмови в обслуговуванні (Denial of Service).
- Пріоритет голосу (FXS) більше високий, чим пріоритет LAN портів.
- Сумісність із Asterisk V1.2, V1.4.

Інтерфейси:

Ethernet (RJ-45, 10/100 base-T)

- 1-WAN порт для підключення до маршрутизатора, ADSL модему (ATU-R) або комутатору.
- 4-LAN порти для підключення комп'ютерів або інших мережних пристроїв.

Телефонні порти (RJ-11)

- 2-FXS порти для підключення аналогових телефонів (POTS).
- 1-PSTN порт для підключення PSTN лінії (backup).

Антенa з RF з'єднувачем

Живлення

- DC 12V, 1.5A.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

Кнопка скидання в заводські установки (Factory Default)

Світлодіодні індикатори:

- WAN, LAN, FXS, PSTN, WLAN і Power.

Маршрутизація (IP шлюз)

- Підтримка IPv4 (RFC 791).
- Підтримка MAC адрес (IEEE 802.3).
- Підтримка TCP/UDP (RFC 793/768).
- Підтримка RTP/RTCP (RFC 1889/1890).
- Підтримка ICMP (RFC 792).
- Підтримка ARP (RFC 826).
- Підтримка статичних IP (WAN).
- Підтримка DHCP Client (RFC 2131).
- Підтримка DHCP Server (RFC 2131).
- Підтримка PPPo Client (RFC 2516).
- Підтримка DNS Client.
- Підтримка DNS Relay.
- Підтримка Dynamic DNS.
- Підтримка NAT/NAPT (RFC 1631).
- Підтримка Virtual DMZ.
- Підтримка Firewall:
 - IP фільтрація.
 - Port Filtering.
 - MAC Filtering.
 - URL Filtering.
 - Port Forwarding.
- Запобігання DoS: Denial of Service.

Брідж

- Підтримка VPN Pass-Through: IPsec, PPTP, L2TP.
- Підтримка NAT Traversal.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

- UPn IGD.
- STUN Client.
- Підтримка SNTP Client (RFC 2030).
- Підтримка HTTP Server.
- Підтримка Quality of Service.
- 802.1Q VLAN Tag.
- DiffServ (RFC 2475).

Бездротові мережі (Wireless LAN)

- Сполучимо зі стандартом IEEE 802.11 b/g.
- Смуга частот: 2.4GHz (B)/2.4GHz (G)/2.4GHz (B+G).

Моди:

- WiFi AP.
- WiFi Client.
- WDS Repeater.
- WiFi AP+WDS Repeater.
- Universal Repeater.
- SSID/Broadcast SSID.
- Канали: 1 ~ 11, 13 для Європи (ETSI), 14 для Японії.
- Тип автентифікації: Open System/Shared Key.
- Швидкість: 1М, 2М, 5.5М, 11М, 6М, 9М, 12М, 18М, 24М, 36М, 48М, 54М.
- Бездротовий ISP (Wireless ISP): Bridged WAN і LAN спільно, WiFi Client, підключений до ISP AP.
- Шифрування й автентифікація 802.1x.
- WEP (64/128bits), WPA/WPA 1-TKIP/AES, Mixed WPA і WPA2.
- WPA Authentication: PSK: Pre-Shared Key (Personal)/Radius (Enterprise).
- Контроль доступу по MAC адресах (дозволений/заборонений).
- Підтримка 802.11f, IAPP (Inter Access Point Protocol).
- Підтримка типів мережі: Infrastructure і Ad-Нос моди.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

- Підтримка автоматичного й ручного сканування з'єднання.
- До 8 WDS AP.

IP телефонія (VoIP)

- Підтримка SIPv2 (RFC 3261).

Проксі-установки для підвищення продуктивності й безпеки:

Голосові кодеки:

- G.711 (a-Law/u-Law): 64 bits (PCM).
- G.723.1: 6.3k/5.3k bits.
- G.726: G.726:16/24/32/40K bits (ADPCM).
- G.729A: 8k bits (CS-SCELP).
- G.729B: з VAD і CNG відповідно до G.729.
- VAD (Визначення пауз), CNG (Генерація шуму в паузах).
- Лунапридушення сумісне зі стандартом G.168/165.
- Динамічний буфер пакетів.
- Інтерполяція збійних пакетів.
- Використання DTMF у вхідним/вихідних з'єднаннях у відповідності сRFC 2833 і SIP.

Підтримка Caller ID:

- Підтримка генерації Caller IDDTMF, FSK (Bellcore, ETSI, BT, NTT).
- Очікування дзвінка з Caller ID.
- Short Ring/Dual Tone перед Caller ID.

Функції факсу:

- Автоматичне виявлення факсу.
- Підтримка G.711 pass-through.
- Підтримка протоколу T.38.

Модемна передача:

- Автоматичне виявлення тону.
- Підтримка G.711 pass-through.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- Системна статистика.
- Збереження /перезавантаження установок.
- Системна балка зберігається локально або дистанційно.
- Установка дати й часу.

Сертифікати

- FCC (Part 15, Class B) і CE.

Фізичні параметри

- Розміри: 175*32*126 mm.
- Вага: 250 гр.

Параметри навколишнього середовища

- Вологість 10-85% (без конденсату).
- Робоча температура від 0 до +45 °С.
- Температура при зберіганні від 0 до + 55 °С.

Живлення

- Зовнішнє джерело живлення.
- вхідна напруга AC100 V-240V, 50-60Hz.
- вихідна напруга – DC 12V, 1.5A.

Що за звір цей SMB?

Спробуємо розібратися, що таке малий і середній бізнес. У цей час існують щонайменше два критерії – чисельність працюючого персоналу й оберт компанії.

Відповідно до рекомендацій Єврокомісії по кількості співробітників, до підприємств малого й середнього бізнесу відносять ті фірми, штат яких не перевищує 50 і 250 чоловік відповідно. У США ж до малого бізнесу ставляться фірми зі штатом 100 чоловік, до середнього – 500 чоловік.

Вирішили, що чисельність працівників повинна бути:

- а) від 101 до 250 чоловік включно для середніх підприємств;
- б) до 100 чоловік включно для малих підприємств;
- в) мікропідприємства – до 15 чоловік.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

З обігом усе ще складніше. Якщо в ЄС для мікро-, малих і середніх підприємств норма – 2, 10 і 43 млн євро, то для України норма – 1,2, 9,2 і 23,2 млн євро відповідно.

Що ж стосується України, то єдиним і не зовсім конкретизованим згадуванням якої-небудь класифікації бізнесу є розділ 1 Закону України «Про державну підтримку малого підприємництва». Відповідно до даної розділу суб'єктами малого підприємництва є:

- фізичні особи, зареєстровані у встановленому законом порядку як суб'єкти підприємницької діяльності;
- юридичні особи – суб'єкти підприємницької діяльності будь-якої організаційно-правової форми й форми власності, у яких середньооблікова чисельність працюючих не перевищує 50 чоловік і обсяг річного валового доходу не перевищує 200 млн грн (близько 6,7 млн євро).

Так що назвати класифікацію неточної не можна, її практично немає. Під середнім бізнесом за рубежом і в нас розуміються значно різні по розмірі підприємства: там середній бізнес – це в основному компанії з оборотом від \$30 до \$100 млн. У нас же підприємство, що має декількох десятків мільйонів гривень річного обороту, упевнено ставиться до середнього бізнесу, так як малий бізнес – це обороти не більше декількох мільйонів гривень.

Загалом, наш середній – це їх дрібний.

Чому це важливо? Для того щоб успішно працювати, клієнта потрібно знати в особу, знати його потреби й можливості. А під описану класифікацію попадає зараз порядку 90 % компаній, що працюють на українському ринку.

Так що раз вуж фахівці не домовилися, давайте ми з вами домовимося, що під SMB ми сьогодні будемо розуміти:

- компанії із чисельністю співробітників до 50 чоловік (і їсти причини не вважати, скільки вони заробляють)
- відособлені офіси до 40-50 чоловік (головний офіс не розглядаємо – це тема іншої розділу, але вважаємо, що вся мережа будується на IP).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

Вибір рішення

Отже, сьогодні ми розглянули кілька рішень по організації зв'язку в невеликому офісі. Як же не заплутатися в ухваленні рішення й виборі того або іншого виробників?

Так як всі системи використовують одну технологію, всі «базові» функції в них практично однакові. Тому при виборі рішення переконаєтеся, що ці функції підтримуються, а саме:

- безкоштовний телефонний зв'язок по внутрішньої телефонної IP-мережі;
- можливість забезпечити голосовим зв'язком співробітників практично на будь-якій відстані (у будь-якій точці миру, де є доступ в Інтернет);
- можливість використання безкоштовних програмних клієнтів (softphone);
- дешевина інсталяції, так як все працює по одному кабелі – немає необхідності прокладати додаткові кабелі;
- значне здешевлення обслуговування мережі. Практично все зводиться до обслуговування IP-мережі, а адміністрування вимагає лише базових знань IP-телефонії.

Визначитеся з тим, що вам необхідно й що ви хочете одержати в результаті.

1. Якщо вам необхідно просте й недороге рішення і якщо ви приблизно знаєте свої необхідні потреби, то варто звернути увагу на постачальників «коробкових» рішень.

2. Якщо ваш бізнес вимагає певної специфіки роботи й додаткових функцій, як то поетапне масштабування системи, організація контакт-центру й т.п., звертайтеся до постачальників більше «великовагових» рішень.

При спілкуванні з постачальником завжди уточнюйте:

1. Чи остаточна це вартість?
2. Чи все устаткування враховане при складанні пропозиції?

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

3. Чи включена в пропозицію вартість всіх необхідних ліцензій для відкриття тої або іншої системи?

4. Немає чи необхідності докупувати додаткове устаткування (наприклад, GSM-шлюзи для інтеграції з мережами мобільного зв'язку)?

Не соромтеся звертатися до постачальників тих або інших рішень. Кожної з них з радістю візьметься вирішити ваше завдання. Зрівняєте запропоновані рішення. У когось вийде дешевше, у когось дорожче, у кого те «більш функціональне», у когось із більшим «запасом міцності», у когось «на виріст». У кожному разі – вибрати вам, відповідно до ваших переваг і вимогам до системи.

Що хочеться сказати на закінчення? Тенденція останнього років говорить про те, що IP -телефонія поступово, але планомірно витісняє з ринку традиційну телефонію. Наскільки незабаром відбудеться ця зміна – сказати важко. Але зробити зв'язок функціональним, доступним, а головне – недорогим для невеликої компанії IP-телефонія може вже сьогодні

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero RAD Studio Delphi 10.3.2 Rio Architect – це найшвидший спосіб створювати й оновлювати інтенсивно працюючі з даними, сильно взаємодіючі застосунки з візуально насиченим користувальницьким інтерфейсом для Windows 10, Mac, мобільних пристроїв, IoT і інших платформ за допомогою Object Pascal і C++. Широкий вибір функцій підтримки Windows 10, у тому числі і нові компоненти VCL для Windows 10, стилі для VCL і FMX, а також служби UWP (універсальної платформи Windows), наприклад повідомлення, дозволяють легко й швидко перенести застосунки в Windows 10, зберігши користувачів. Нова платформа дозволяє підтримувати великі проекти на більшому числі платформ із подвоєним обсягом пам'яті в середовищі розробки й удвічі більшим розміром підтримуваних проектів. Крім того, підтримка декількох моніторів і десятки

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

– Темне й світле оформлення Незалежно від того чи волієте ви кодувати вночі або у світлий час доби, завдяки темному й світлому оформленню RAD Studio ви можете вибрати потрібний вам стиль. Було доведено, що темне оформлення допомагає знизити зорову напругу в умовах низького освітлення, дозволяючи вам працювати більш продуктивно вночі. Немає нічого простіше, ніж перейти від темного до світлого оформлення й навпаки за допомогою меню панелі інструментів.

– Виконаєте користувальницьке налаштування свого середовища розробки Поліпшена програма установки інтерфейсу користувача й менеджера ліцензій інтерфейсу користувача дозволяє визначити ті можливості, які необхідні й опустити непотрібні, незалежно від того чи розробляєте ви застосунки для декількох платформ або всього однієї.

– Чистий, оновлений інтерфейс користувача інтегрованого середовища розробки Знайдіть потрібні можливості. Швидко. Головне вікно інтегрованого середовища розробки відцентровано й відрізняється високим ступенем читаності. Ви з легкістю визначите, де перебуває область фокусування клавіатури з оновленими змінами фонових квітів фокуса. Вкладки редактора більше, що полегшує читання шрифтів, тому ви можете швидко внести зміни й зберегти кодування.

– Чудові застосунки Windows з VCL. Бібліотека візуальних компонентів (Visual Component Library, VCL) пропонує просту й візуальну розробку користувальницького інтерфейсу застосунки, у версії 10.3 представлені нові відновлення, які дозволять вашим застосункам виглядати сучасними й свіжими.

– Розширена підтримка HighDPI. Завдяки новому елементу керування VCL High DPI ImageList у версії 10.3 розроблювачі, що створюють нові застосунки VCL для Windows або оновлюючи існуючі застосунки для High DPI дисплеїв, можуть повністю підтримувати зроблені до рівня пікселів зображення зі змінною розв'язною здатністю на всіх елементах керування, а також будь-яке

						ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			46

- 2X швидкість математичної продуктивності для Win64.
- Нові додаткові лабораторії C++ в GetIt.
- Нові й поліпшені можливості роботи з базами даних. InterBase 2017 / IBToGo 2017 в RAD Studio. Версії Professional включають ліцензію розроблювача InterBase 2017, у той час як версії Enterprise і Architect містять у собі ліцензії InterBase ToGo. InterBase ToGo доповнена можливістю шифрування, функціями зміни подань, призначених для простої синхронізації даних застосунку по підписці без обмежень за розміром файлу бази даних.
- Поліпшена й оновлена підтримка для популярних баз даних, включаючи MySQL v8.0, MariaDB 10.3, SQL Server 2017, PostgreSQL v10, Firebird v3.0, MongoDB, InterBase, SQLite 3.23.1, SQL Anywhere і багатьох інших.
- Удосконалення DataSnap.
- Поліпшення REST. Підтримка додаткових родинних REST методів, типів і властивостей.
- Повністю оновлений модуль живлення версії Architect. Одержіть більше від версії Architect, включаючи ці ліцензії сімейства Idera.
- Ліцензія Sencha ExtJS Professional: Створіть свій ідеальний мережний вхідний інтерфейс за допомогою javascript і ExtJS.
- Ліцензія на розгортання InterBase ToGo. Додайте сховище даних у свої застосунки за допомогою цієї гнучкої, зашифрованої бази даних, що вбудовується.
- Ліцензія для розподіленого розгортання RAD Server. Ідеально підходить для серверного застосунку архітектури мікросервісів.
- Ліцензія AquaData Studio. Вражаючий аналіз бази даних.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

забезпечення, яке призначено для системи VoIP для моделі SaaS.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

IP-телефонія – це технологія, що зв'язує воедино переваги телефонії й Інтернет. Донедавна мережі з комутацією каналів (телефонні мережі) і мережі з комутацією пакетів (IP-мережі) існували практично незалежно друг від друга й використовувалися для різних цілей. Телефонні мережі використовувалися тільки для передачі голосової інформації, а IP-мережі – для передачі даних. Технологія IP-телефонії поєднує ці мережі за допомогою пристрою, названого шлюз або gateway. Шлюз являє собою пристрій, у який з однієї сторони включаються телефонні лінії, а з іншого боку – IP-мережа (наприклад, Інтернет).

Загалом передача голосу в IP-мережі відбувається в такий спосіб. Вхідний дзвінок і сигнальна інформація з телефонної мережі передаються на прикордонний мережний пристрій, названий телефонним шлюзом, і обробляються спеціальною картою пристрою голосового обслуговування. Шлюз, використовуючи керуючі протоколи сімейства H.323, перенаправляє сигнальну інформацію іншому шлюзу, що перебуває на прийомній стороні IP-мережі. Прийомний шлюз забезпечує передачу сигнальної інформації на прийомне телефонне устаткування відповідно до плану номерів, гарантуючи наскрізне з'єднання. Після встановлення з'єднання голос на вхідному мережному пристрої оцифровується (якщо він не був цифровим), кодується у відповідності зі стандартними алгоритмами ІТУ, такими як G.711 або G.729, стискується, інкапсулюється в пакети й відправляється по призначенню на віддалений пристрій з використанням стека протоколів TCP/IP. Приходжі на прийомний шлюз IP-пакети перетворюються назад у телефонний сигнал і приймаючого абонента одержує виклик. Кінцеві споживачі послуги можуть навіть не догадуватися про те, як здійснюється цей дзвінок.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

невелике клієнтське програмне забезпечення, що звичайно можна знайти на тій же Web-сторінці, і яке встановлюється автоматично. З іншої сторони Surf&Call дозволяє представникам компаній відповідати на питання, демонструвати Web-сторінки, передавати необхідну інформацію, поліпшуючи тим самим якість надаваних послуг.

Переваги IP-телефонії

Здешевлення телефонних переговорів. Впровадження технології VOIP у рамках обчислювальної мережі дозволяє зменшити сумарні витрати, пов'язані з веденням міжнародних і міжміських телефонних переговорів, а також почати процес міграції до технологій пакетної передачі мультимедійних даних. Крім того, з огляду на можливість виходу на міську телефонну мережу, використання цієї технології може звести до мінімуму оренду звичайних телефонних ліній.

Поліпшена якість зв'язку. Якість зв'язку можна оцінити, використовуючи наступні основні характеристики: рівень перекручування голосу; частота «провалля» голосових пакетів; час затримки (між проголошенням фрази першого абонента й моментом, коли вона буде почута другим абонентом). По всіх перерахованих характеристиках якість зв'язку значно збільшилося в порівнянні з першими версіями рішень IP-телефонії, які допускали перекручування й переривання мови. Поліпшення кодування голосу й відновлення загублених пакетів дозволило досягти рівня, коли мова розуміється абонентами настільки добре, що співрозмовники не догадуються, що з'єднання відбувається за технологією IP-телефонії. Зрозуміло, що затримки впливають на темп бесіди. Відомо, що для людини затримка до 250 мільсекунд фактично непомітна. Існуючі на сьогоднішній день рішення IP-телефонії не перевищують ця межа, так що розмова фактично не відрізняється від зв'язку по звичайній телефонній мережі. Крім цього, затримки зменшуються завдяки наступним трьом факторам:

– По-перше, удосконалюються телефонні сервери (їхні розроблювачі борються із затримками, поліпшуючи алгоритми роботи).

						ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			53

– По-друге, розвиваються частки (корпоративні) мережі (їхні власники можуть контролювати ширину смуги пропускання й, отже, величини затримки).

– По-третє, розвивається сама мережа Інтернет – сучасний Інтернет не був розрахований на комунікації в режимі реального часу. The Internet Engineering Task Force (IETF) разом з операторами мереж Інтернет пропонують нові технології, такі, як Reservation Protocol (RSVP), які дозволяють резервувати смугу пропускання.

Рішення проблеми зайнятої лінії. Уже давно аматори бороздити всесвітню мережу зіштовхуються із проблемою зайнятості телефонних ліній під час сеансу Dial-up. IP-телефонія дозволяє дуже елегантно вирішити цю проблему. Єдине, що повинен зробити абонент – це замовити на своїй АТМ переадресацію по сигналі «зайнято» на телефонний номер сервера IP-телефонії. При дзвінку на номер абонента під час Інтернет-сесії виклик переадресується на сервер IP-телефонії, що перетворює його в IP -пакети й відправляє на комп'ютер абонента. На комп'ютері абонента з'являється іконка «Вхідний дзвінок», кликнувши на яку він може поговорити із що дзвонить.

Підвищення якості факсимільного зв'язка. Так як, по суті факсимільне повідомлення – потік цифрових даних, а в технології VoIP дані передаються в цифровому виді, тому передача факсимільних повідомлень по аналогових лініях скорочується до мінімуму. А за рахунок того, що устаткування має можливість демодулювати сигнал перед передачею по IP-мережі й передавати закодоване в 64 Кбітному форматі факс-повідомлення в смузі 9,6 Кбіт, знижується навантаження на канали.

Інтеграція філій у єдину інформаційну структуру. Останнім часом з розвитком інформаційних технологій і збільшенням пропускну здатності каналів усе для найбільш оперативного рішення ділових завдань філії компанії поєднують в одне ціле, створюючи інтрамережу. Так як пропонується технологія використовує для передачі голосу саме мережі передачі даних, то з'являється можливість поєднувати не тільки комп'ютерні мережі, але й телефонні.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Віртуальні приватні мережі (VPN). IP-телефонія є ідеальною технологією для побудови віртуальних приватних мереж підприємства. Головна риса технології VPN – використання IP-мережі як магістраль для передачі корпоративного IP-трафіку. Мережі VPN вирішують завдання підключення корпоративного користувача до віддаленої мережі й з'єднання декількох віддалених ЛОМ і АТМ у єдину корпоративну мережу передачі голосу й даних.

Глобальний роумінг. IP-телефонія дозволяє операторам зв'язку дуже просто й з мінімальними витратами організувати роумінг послуг зв'язку. Це особливо актуально для операторів мобільного зв'язку – рішення, побудоване на технологіях IP-телефонії, на порядок дешевше традиційного, і має набагато більшу гнучкість.

Сполучений доступ в Інтернет. Голосові дані, факсимільні повідомлення передаються з використанням IP – основного набору протоколів Інтернет, дане рішення саме собою має на увазі доступ до ресурсів Мережі й очевидна економія на оренду ліній зв'язку й оплату послуг.

Мінімальні вкладення в устаткування. Якщо Ви використовуєте устаткування Cisco Systems, що відповідає всім сучасним стандартам, то Вам не буде потрібно прибігати до яким або витратам на додаткове устаткування, як телефонне так і комутаційне. Крім того, продукція Cisco має достатню гнучкість і масштабованість, тобто нарощувати потужність, продуктивність і функціональні можливості можна поступово, відповідаючи потребам, що розвиваються.

Устаткування для IP-телефонії

Компанія Cisco Systems у цей час є лідером на ринку систем IP-телефонії, так як близько 92% трафіку VoIP передається по її устаткуванню. Недавно фірма оприлюднила власну стратегію розвитку пакетної телефонії, у рамках якої планується реалізація технологій VoIP, VoFR і навіть VoIP over ATM, що забезпечує якість голосу на рівні вимог, пропонованих до устаткування телекомунікаційних операторів (carrier-class quality). Компанія пропонує комплексний підхід до створення архітектурних систем з інтеграцією голосу,

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

відео й даних Cisco AVVID (Arc hitecture for Voice, Video and Integrated Data), що включають мережну інфраструктуру, клієнтські місця, серверні й користувальницькі додатки.

Рішення Cisco IP-телефонії складається з наступних компонентів: спеціалізовані цифрові IP-телефони (серії 7960/7940, 7910/7910+SW); керуючий сервер Cisco CallManager (на основі серверів MCS 7815-1000, MCS 7825-1133, MCS 7835-1266); голосові шлюзи для стикування IP-мереж з телефонною мережею загального користування (моделі Cisco 1750, 1760, VG200, 2600, 3600, 3700, Cisco 7200, AS5350, шлюзові модулі Catalyst 6000/6500, 4000), а також користувальницькі голосові додатки. Область застосування пропонованих VoIP-рішень охоплює користувальницьке прикінцеве устаткування (CPE), платформи доступу, що комутирується, виробу для ліній xDSL, кабельні інфраструктури й системи виділеного доступу, що забезпечують повну обробку трафіку.

Майбутнє IP-телефонії

У своєму розвитку IP-телефонія пройшла три етапи. На першому це була, скоріше, Internet-іграшка, придатна тільки для що квакає й сичить для зв'язку двох ентузіастів. Два комп'ютери, оснащені мікрофонами, динаміками, звуковими картами й не дуже складним програмним забезпеченням, дозволяли вести двосторонній діалог через Internet у реальному часі. Однак до зручностей звичайної телефонної послуги такий спосіб спілкування явно недотягав: абонентам потрібно було знати IP-адреса комп'ютера співрозмовника, домовлятися про час розмови, вибирати момент для більше якісної передачі мови, коли трафік Internet не зіштовхувався з більшими перевантаженнями й затримками. Крім того, при відсутності стандартів на обох комп'ютерах було потрібно встановити таке програмне забезпечення, щоб спосіб кодування голосу й упакування його в пакети був тим самим. Взаємодія між комп'ютером і телефоном, підключеним до звичайної телефонної мережі, не передбачалося. Зате витрати обмежувалися невеликою платою провайдеру Internet.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

Другий етап ознаменувався появою стандартів IP-телефонії, насамперед – стандартів групи H.323. Розроблювачі цих протоколів виходили з того, що дві мережі – телефонна й IP – будуть співіснувати пліч-о-пліч досить тривалий час, а виходить, важливо регламентувати їхнє взаємодія з обліком існуючих у традиційних телефонних мережах процедур устанавлення з'єднання, а також домовитися про спосіб передачі виклику й самого голосу по мережі IP. У стандартах H.323 визначається дві групи протоколів – протоколи транспортної площини (transport plane), називаною також користувальницькою площиною (user plane), і протоколи площини керування викликами (call control plane). На цьому етапі розвитку IP-телефонії мережа IP (Internet або приватна) широко використовувалася в якості транзитної між двома місцевими телефонними мережами. Дана схема реалізації загальнодоступних послуг IP-телефонії стала досить популярна в усьому світі. Для її реалізації операторів зв'язку не треба створювати власну дорогу транспортну інфраструктуру й мати безпосередній доступ до абонентів. Однак стратегічні перспективи такого підходу – залишають бажати кращого через невисокий ступінь масштабованості й вузького спектра послуг.

Масштабованість обмежується декількома факторами. По-перше, провайдеру доводиться встановлювати численні однорангові зв'язки зі своїми друзями-суперниками по бізнесі. По-друге, протоколи обох площин необхідно реалізовувати у всіх елементах мережі IP-телефонії: і у воротарях, і в шлюзах, і в терміналах, що приведе до зайвої складності й дорожнечі всіх цих пристроїв. І, нарешті, користувачам надаються тільки базові послуги з обробки викликів, оскільки взаємодія із протоколами міжстанційної сигналізації SS7 і з послугами інтелектуальної мережі IN відсутній.

Крім того, діалог із сервером інтерактивної голосової відповіді при автентифікації абонента й завданні номера викликуваного абонента досить стомлюючий – набагато зручніше просто набрати цей номер з невеликою приставкою начебто 8-20 і одержати доступ до послуг міжнародної IP-телефонії. Але для цього провайдеру потрібний прямий доступ до абонента або

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

домовленість із місцевими операторами про переадресацію таких викликів на шлюз ІРТР за допомогою засобів інтелектуальної мережі (а вони поки підтримуються далеко не всіма місцевими операторами).

Таким чином, для виходу ІР-телефонії на більше високий рівень національного або міжнародного оператора потрібні інші стандарти й устаткування, щоб мережі, побудовані на базі протоколу ІР, могли рівноправно сусідити із традиційними телефонними мережами.

Багато хто з необхідних стандартів уже з'явилися й втілені в новому поколінні устаткування, що служить основою для третього етапу розвитку ІР-телефонії. Така мережа може підтримувати власних абонентів і служити транзитною мережею для традиційних телефонних мереж з наданням повного спектра послуг, включаючи послуги інтелектуальної мережі ІN. У вузлах ІР-телефонії нового покоління відбувся чіткий поділ функцій на три групи – транспортну, керування викликами й прикладними сервісами. На цьому етапі підтримується весь спектр додаткових послуг, які можуть надавати для абонентів розвинені телефонні комутатори міського типу, у тому числі й за допомогою інтелектуальної мережі: переадресацію викликів відповідно до різних умов, телеголосування, безкоштовний дзвінок, дзвінок по спеціальному тарифі, скорочений набір і т.п.

Дуже важливо, що взаємодія між рівнями здійснюється через стандартні інтерфейси, а це створює серйозні передумови для побудови телефонних вузлів ІР-телефонії на основі продуктів різних виробників із застосуванням загальноприйнятих способів обробки викликів.

Очевидно, що в ІР -телефонії є майбутнє, але в якій послідовності й коли буде здійснюватися широкомасштабний перехід до неї поки невідомо, та й загальна криза телекомунікаційної галузі цьому явно не сприяє. Проте будемо сподіватися, що потенціал ІР-телефонії, що набрав за багатьма ознаками достатню критичну масу, буде незабаром реалізований, і рядові користувачі й численні провайдери зможуть скористатися всіма її перевагами.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

3.2 Розробка структурної схеми

Провідна телефонія за своє більш ніж віковий розвиток у силу технічних і економічних перешкод не змогла стати загальним надбанням людства. У розвинених країнах телефонні апарати були встановлені у всіх держустановах, комерційних фірмах, практично в кожній міській квартирі й сільському будинку, а от у нас стаціонарний телефонний зв'язок навіть наприкінці ХХ сторіччя залишалася для багатьох недосяжною не тільки у віддалених і малонаселених районах, але й у великих селищах. Але ж людям для оперативного рішення безлічі особистих і бізнес-питань необхідно мати можливість голосового спілкування «тут і зараз».

Мобільна телефонія, що почала свою експансію біля сорока років тому, значно підвищила доступність голосового зв'язку, особливо на неохоплених стаціонарним зв'язком територіях, і за короткий час стала технологією масового застосування.

Провідна й бездротова технології освоїлися у своїх нішах і благополучно співіснували, не турбуємі конкурентами, ще яких-небудь десять років тому, коли раптом з'явився третій «суперник» – VoIP-технологія. І суперник цей виявився небезпечний, насамперед, тим, що надав можливість істотно меншої оплати за ті ж мінути голосового зв'язку, які могли б бути витрачені в мережах його попередників. Причому він може легко «впроваджуватися» і в стаціонарні, і в мобільні зони, здійснюючи зв'язок з абонентами й першої, і другої технології мовного спілкування. Не говорячи вже про «власний контингент на власній VoIP-території».

Технологія VoIP для моделі SaaS: чому дешевше?

Для надання послуг стаціонарної телефонії необхідно було створити інфраструктуру у вигляді кабельних ліній зв'язку величезної довжини. Для мобільної – побудувати базові станції передачі радіосигналу. А фізичним середовищем для технології VoIP для моделі SaaS став Інтернет, тобто

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

провайдером нового телекомунікаційного напрямку не довелося створювати свою інфраструктуру: є Інтернет – є можливість пропозиції сервісів VoIP для моделі SaaS, що й відбито в «ім'ї» технології – Voice over Internet Protocol – «голос по інтернет-протоколу».

Витрати на будівництво базових станцій вертаються операторові в оплаті за розмови по мобільних телефонах. Провайдери VoIP-технології, що впровадилися в «готову» мережу Інтернет, таких витрат на VoIP для моделі SaaS устаткування не несли, і, відповідно, у тарифах за послуги зв'язку вони не присутні.

Правда, назва VoIP для моделі SaaS не повною мірою характеризує технологію, що дозволяє здійснювати прийнятно-передачу не тільки мови (цьому сегменту привласнена ім'я «IP телефонія»), але й відеоконтенту й, взагалі, будь-яких даних, представлених у цифровому виді. Однак оскільки сьогодні найбільш затребувана саме IP-телефонія, то на її прикладі ми й розглянемо всі основні техніко-економічні показники VoIP для моделі SaaS, попутно відзначивши, що ця технологія крім роботи в Інтернеті може бути реалізована в будь-яких виділених цифрових каналах, що підтримують інтернет-протокол і складових IP-мережа.

Отже, перший доданок економічності IP-телефонії (вірніше було б, напевно, сказати – «від'ємник») – це відсутність інвестицій у створення інфраструктури, які в операторів стаціонарного й мобільного зв'язку повинні окупитися, для чого вони «незримо присутні» у їхніх тарифах.

Друга обставина, що дозволила провайдерам IP-телефонії встановити мінімальну планку оплати їхніх послуг, полягає в тім, що в телефонних мережах загального користування (ТМЗК, ТфЗК), оплата розмови визначається його тривалістю й довжиною виділеного каналу. А в IP-телефонії оплачується лише підключення до Інтернету й обсяг переданого трафіку.

Третя стаття витрат провайдерів стаціонарного зв'язку, що закладається в тарифи, як це ні парадоксально звучить, – оплата пауз у розмовах. Справа в тому,

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

телефонів у довільних комбінаціях. Відзначимо, що звичний нам аналоговий телефонний апарат можна перетворити в IP-телефон, включивши між його входом і інтернет-розеткою аналоговий телефонний адаптер (VoIP АТА).

Кожний IP-телефон (із провідним і бездротовим входом/виходом або убудованим аналоговим модемом) підключається до Мережі інтернет-провайдером, потім проходить реєстрацію в оператора послуг IP-телефонії, одержуючи при цьому логін і пароль.

Але для використання цієї технології наявність спеціального апарата необов'язково, якщо встановити на комп'ютері програму-клієнт, що імітує телефон, і підключити до ПК навушники й мікрофон або USB-телефон, що виконує функції навушників і мікрофона. Такий комп'ютерно-програмний комплекс називається софтфоном (програмним телефоном). ПЗ SoftPhone можна безкоштовно скачати на web-сайті провайдеру IP-телефонії, після чого там же зареєструвати свій пристрій. Самою популярною безкоштовною програмою для VoIP для моделі SaaS телефонії у світі є Skype.

Різновидом програмних телефонів є двірежимні GSM/WiFi (стільниковий/VoIP) мобільні телефони, використовувані у двох «іпостасях»: в GSM-мережі вони поведуться як стільниковий телефон, а в WiFi-зоні (при встановленому в телефоні ПЗ SoftPhone) – в IP-мережі. Причому в другому режимі роумінг у мобільній мережі практично безкоштовний.

Відеотелефон – устаткування, що працює по VoIP для моделі SaaS технології, забезпечує ефект присутності в офісі абонента, що перебуває «на іншому кінці проведення», завдяки чому підвищується результативність бізнес-переговорів.

Для організації голосового зв'язку, здійснюваної між комп'ютерами, IP-телефонами й відеотелефонами, досить з'єднати їхнім кабелем з інтернет-розеткою й увійти в Інтернет. А для зв'язку IP-мережі із ТМЗК необхідне застосування аналогових VoIP для моделі SaaS шлюзів FXS або FXO. У випадку ж цифрових телефонних мереж ISDN (Integrated Services Digital Network) їхній зв'язок із Всесвітньою павутиною забезпечується цифровими VoIP-шлюзами.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

VoIP-шлюзи дозволяють підключитися до мереж декількох операторів для створення декількох маршрутів трафіку з мінімальними тарифами, зарезервувати їх для використання при виникненні перевантажень і відмов у мережах стаціонарного й мобільного зв'язку. При цьому, завдяки наявності альтернативних маршрутів, компанії можуть досягти помітного зменшення витрат на послуги зв'язку, збільшивши до того ж доступність абонентів, що перебувають у мережах, які підтримують різні технології передачі голосу.

В VoIP-шлюзах можуть бути створені віртуальні об'єкти, що визначають маршрутизацію телефонних дзвінків, що дозволяє компанії «безмежно» підключати прямі номери в будь-якій державі планети (ця технологія називається Direct Inward Dialing, DID).

VoIP-шлюзи – міжмережеве устаткування VoIP для моделі SaaS для переведення голосового трафіку між мережами традиційної телефонії й мережею передачі даних.

Ще одне достоїнство IP-телефонії: можливість простого нарощування в офісі кількості номерів (так званої «номерної ємності»), що, як правило, на превелику силу вдається зробити в ГфЗК (і те, якщо в оператора стаціонарного зв'язку є технічна можливість).

Як ми вже відзначали на початку розділу, привабливість IP-телефонії для масового користувача полягає в економії оплати за голосовий зв'язок у порівнянні з оплатою за такі ж переговори в стаціонарних і мобільних мережах. Цей показник, звичайно, важливий і бізнес-сегменту, для якого, виявляється, є ще один «бонус» – низькі витрати на створення корпоративної мережі IP-телефонії (апаратна IP АТМ, програмні АТМ, віртуальні АТМ) на основі існуючої УАТМ.

Підводні камені технології VoIP для моделі SaaS

Ви вже наслухані про переваги й вигоди IP-телефонії й навіть неодноразово бачили устаткування й програми для VoIP для моделі SaaS у роботі в когось зі своїх друзів. Сподобалося. І от ви вирішуєте впровадити її в себе. Але чи розповіли вам про парочку «підводних каменів», які можуть звести «на ні» всі ваші очікування? А вони є.

						ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			63

Відома приказка, що затверджує, що «недоліки є продовження наших достоїнств», може бути віднесена до усім без винятку технологіям. І, звичайно ж, до IP-телефонії. У цій технології «кульгають» якість передачі голосу й безпека. Але ці ущербності можна усунути, якщо розібратися в їхніх коріннях.

Отже, у ТфЗК якість мови не викликало дорікань тому, що в цій технології, заснованої на комутації каналів зв'язку, «пісня ллється» безупинно в одній і тої ж «трубі». А VoIP -технологія передає ту ж «пісню» вроздріб (у вільні тепер каналах) відповідно до інтернет-протоколу, здійснюючи комутацію пакетів даних (як це прийнято в передачі будь-якої інформації з Інтернету). І якщо, наприклад, у передачі текстів і фотознімків, ця пакетна послідовність може закінчитися «коли вийде», і ми спокійно почекаємо її закінчення й відкриємо «готовий добуток», те голосовий зв'язок працює в режимі реального часу й не чекає, коли прийде наступний пакет.

Тому якщо час затримки в одержанні пакетів і їхня пропажа перевищують установлені стандартом величини, це неминуче позначається на розбірливості, чистоті, рівні гучності, появі луни, хрипів і інших некомфорних для слухового сприйняття звуків. Але, як говориться в популярному мультфільмі, «неприємність цю ми переживемо». Природно, за допомогою системного інтегратора, що зможе зробити всі роботи для інтелектуальної обробки затримок одержання пакетів, інтерполяції (часткового відновлення) інформації, що перебувала в загублених пакетах, лунапридушення й керування рівнем голосового сигналу. І буде всі так само добре, як вам дозволив побачити на своїй фірмі ваш друг.

З безпекою – складніше, адже в програму вашого візиту на фірму друга «за замовчуванням» не входила презентація на цю тему. Напевно, вам розповідали, що розмови по IP-телефонії простіше прослуховувати, чим при використанні звичайного стаціонарного апарата. І ці слухи небезпідставні, оскільки для втручання в розмову, що відбувається по ТфЗК, необхідно фізичне підключення до лінії, по якій ви ведете бесіду. А це досить важко (хоча й можливо).

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Якщо ж ви користуєтеся послугами IP-мережі, то досвідчений хакер без особливої праці може «обчислити» вашу IP-адресу й непомітно (програмним шляхом) влізти у ваші розмови, не тільки підслухуючи їх, але й блокуючи й навіть коректуючи. І тут, щоб не вдаватися в зайві для читача подробиці, можна сказати, що фахівці з інсталяції VoIP-технології, здатні створити IP-мережа, рівень захисту якої буде вище, ніж у традиційної провідної телефонної мережі.

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленого програмного забезпечення.

З функціональної схеми ми бачимо, що основний функціональний модуль включає в себе наступні функціональні підблоки:

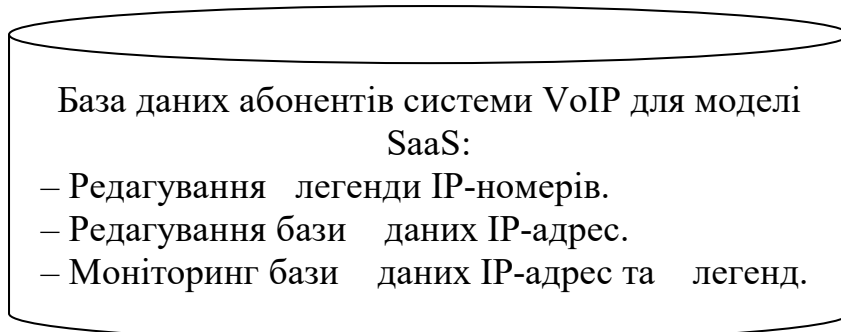
- База даних абонентів системи VoIP для моделі SaaS.
- Білінг.
- Виведення статусу ПЗ.
- IP-телефонія.
- Селективний зв'язок.
- Відеоконференція.
- Формування вхідної інформації.
- Довідник.

Дані, які поступають з основного функціонального модуля перетворюються згідно протоколу H.323, після чого передаються до SaaS.

Робота з програмою починається з введення інформаційного вікна й активізації системи меню. Робота програми здійснюється по діалоговому і дійному режиму, при цьому по діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна..

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Основний функціональний модуль



IP-телефонія

Селективний зв'язок

Відеоконференція

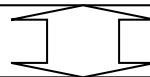
Формування вхідної інформації

Виведення статусу ПЗ

Біллінг

Довідник:

– Довідкова система. – Інструкція користувача.



H.323

Керування з'єднанням і сигналізація: H.225.0, H.225.0/RAS, H.245

Обробка звукових сигналів: G.711, G.722, G.723.1, G.728, G.729

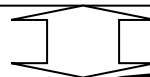
Обробка відеосигналів: H.261, H.263

Конференц-зв'язок для передачі даних: T.120, T.123, T.124, T.125

Мультимедійна передача: RTP, RTCP

Забезпечення безпеки: H.235

Додаткові послуги: H.450



SaaS

Рисунок 3.2– Функціональна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

66

– Поле введення у формі може представляти список допустимих значень, з яких користувачі можуть легко вибрати потрібне.

– Область форми може виводити шаблон, що відповідає поточної виведеної у формі запису.

– Командні кнопки у формі можуть виконувати дії, зв'язані з виведеної у формі поточною записом.

Біллінгова система – автоматизована система розрахунків з абонентами за надані послуги. Керування й статистика може бути доступна з будь-якої точки мережі через Веб-інтерфейс.

Стандарт H.323 визначає широкі вимоги для багатьох різних протоколів, які становлять повний стек протоколів H.323.

Стек H.323 складають 7 груп протоколів:

- керування й сигналізація;
- обробка звукових сигналів;
- обробка відеосигналів;
- конференц-зв'язок;
- передача мультимедійної інформації;
- забезпечення інформаційної безпеки;
- додаткові послуги;

1. Керування з'єднанням і сигналізація:

– H.225.0: протоколи сигналізації й пакетування мультимедійного потоку (використовує підмножину протоколу сигналізації Q.931).

– H.225.0/RAS: процедури реєстрації, допуску й стану.

– H.245: протокол керування для мультимедіа.

2. Обробка звукових сигналів:

– G.711: імпульсно-кодова модуляція тональних частот.

– G.722: кодування звукового сигналу 7 кГц в 64 кбіт/с.

– G.723.1: мовні кодери на дві швидкості передачі для організації мультимедійного зв'язку зі швидкістю передачі 5.3 і 6.3 кбіт/с.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

– G.728: кодування мовних сигналів 16 кбіт/с за допомогою лінійного проорокування з кодуванням сигналу порушення з малою затримкою.

– G.729: кодування мовних сигналів 8 кбіт/с за допомогою лінійного проорокування з алгебраїчним кодуванням сигналу порушення сполученої структури.

3. Обробка відеосигналів:

– H.261: відеокодеки для аудіовізуальних послуг зі швидкістю 64 кбіт/с.

– H.263: кодування відеосигналу для передачі з малою швидкістю.

4. Конференц-зв'язок для передачі даних:

– T.120: це стек протоколів (який включає T.123, T.124, T.125) для передачі даних між окінцевими пунктами. Він може використовуватися для різних додатків в області спільної роботи (Collaboration Work), такий як колективне редагування растрових зображень, спільне використання додатків і спільна організація документів. В T.120 застосовується багаторівнева архітектура, подібна моделі OSI.

5. Мультимедійна передача:

– RTP: транспортний протокол реального часу.

– RTCP: протокол керування передачею в реальному часі.

6. Забезпечення безпеки:

– H.235: забезпечення безпеки й шифрування для мультимедійних терміналів мережі H.323.

7. Додаткові послуги:

– H.450.1: узагальнені функції для керування додатковими послугами в H.323.

– H.450.2: переклад з'єднання на телефонний номер третього абонента.

– H.450.3: переадресація виклику.

– H.450.4: утримання виклику.

– H.450.5: паркування виклику (park) і відповідь на виклик (pick up).

– H.450.6: повідомлення про виклик, що надійшов, у стані розмови.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

- Н.450.7: індикація повідомлення, що очікує.
- Н.450.8: служба ідентифікації імен.
- Н.450.9: служба завершення з'єднання для мереж Н.323.

Рекомендації Н.323 передбачають:

- Керування смугою пропусення.
- Можливість взаємодії мереж.
- Платформну незалежність.
- Підтримку багатоточечних конференцій.
- Підтримку багатоадресної передачі.
- Стандарти для кодеків.
- Підтримку групової адресації .

Передача аудіо- і відеоінформації досить інтенсивно навантажує канали зв'язку, і, якщо не стежити за ростом цього навантаження, працездатність критично важливих мережних сервісів може бути порушена. Тому рекомендації Н.323 передбачають керування смугою пропусення. Можна обмежити як число одночасних з'єднань, так і сумарну смугу пропусення для всіх додатків Н.323. Ці обмеження допомагають зберегти необхідні ресурси для роботи інших мережних додатків. Кожний термінал Н.323 може управляти своєю смугою пропусення в конкретній сесії конференції.

Рекомендації Н.323 пропонують засіб з'єднання учасників відеоконференції в різнорідних мережах (наприклад, IP і ISDN, IP і PSTN). Н.323 не прив'язаний ні до яких технологічних рішень, пов'язаним з устаткуванням або програмним забезпеченням. Взаємодіючі між собою додатки можуть створюватися на основі різних платформ, з різними операційними системами. Рекомендації Н.323 дозволяють організувати конференцію із трьома або більше учасниками. Багатоточечні конференції можуть проводитися як з використанням центрального MCU (пристрою багатоточечної конференції), так і без нього. Н.323 підтримує багатоадресну передачу в багатоточечній конференції, якщо мережа підтримує протокол керування груповою адресацією (такий, як IGMP). При

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

багатоадресній передачі один пакет інформації відправляється всім необхідним адресатам без зайвого дублювання. Багатоадресна передача використовує смугу пропускання набагато більш ефективно, оскільки всім адресатам – учасникам списку розсилання відправляється рівно один потік.

H.323 установлює стандарти для кодування й декодування аудіо- і відеопотоків з метою забезпечення сумісності встаткування різних виробників. Разом з тим стандарт досить гнучкий. Існують вимоги, виконання яких обов'язково, і існують опціональні можливості, у випадку використання яких також необхідно строго дотримуватися стандарту. Крім цього, виробник може включати в мультимедійні продукти й додатки додаткові можливості, якщо вони не суперечать обов'язковим і опціональним вимогам стандарту.

Учасники конференції хочуть спілкуватися один з одним, не піклуючись про питання сумісності між собою. Рекомендації H.323 підтримують з'ясування загальних можливостей устаткування кінцевих користувачів і встановлюють найкращі із загальних для учасників конференції протоколів кодування, виклику й керування.

H.323 конференція може включати учасників, кінцеве встаткування яких має різні можливості. Наприклад, один з учасників може використовувати термінал як тільки з аудіо- можливостями, у той час як інші учасники конференції можуть мати можливості передачі/прийому також відео й даних.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

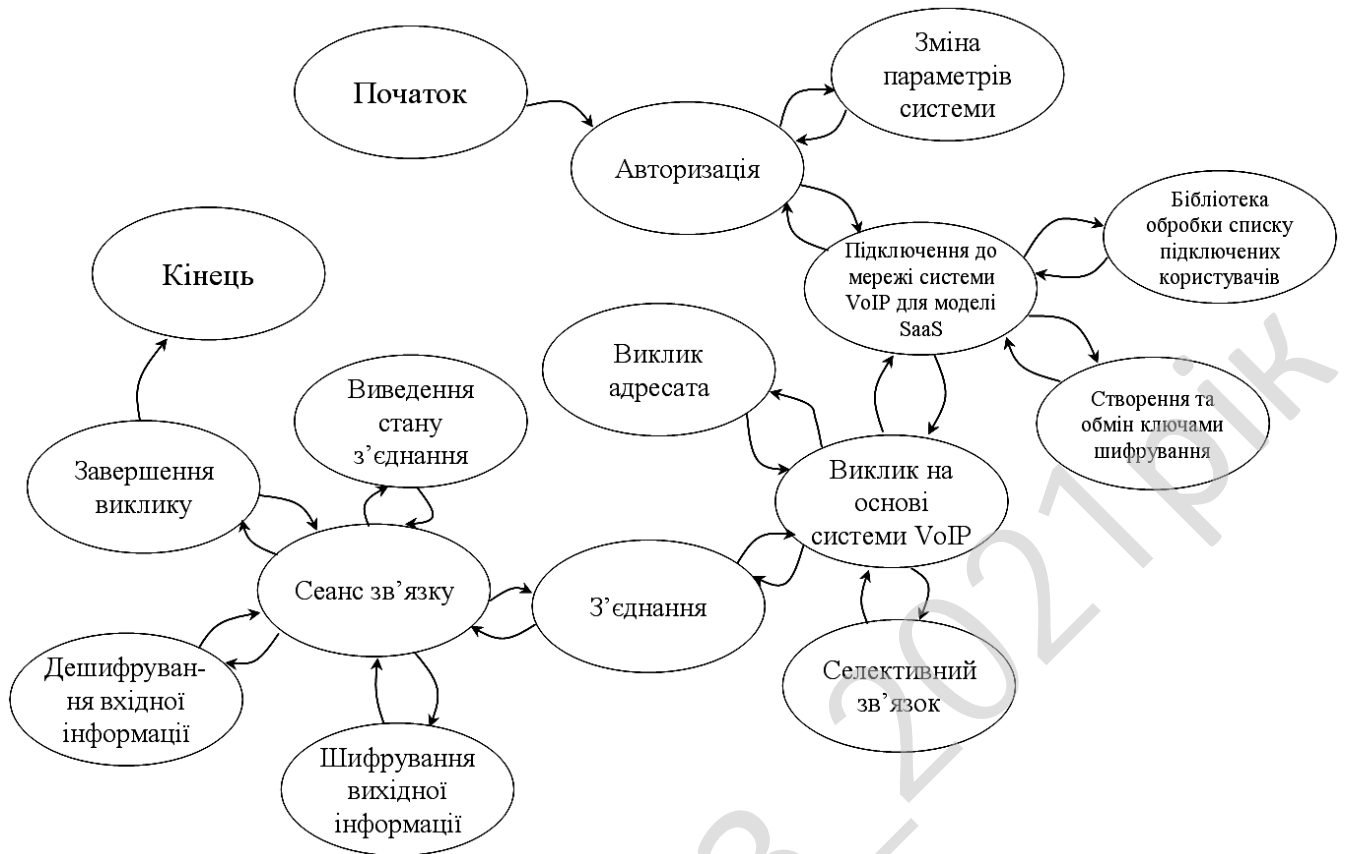


Рисунок 3.3 – Діаграма взаємодії процесів

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію магістерської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми. З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

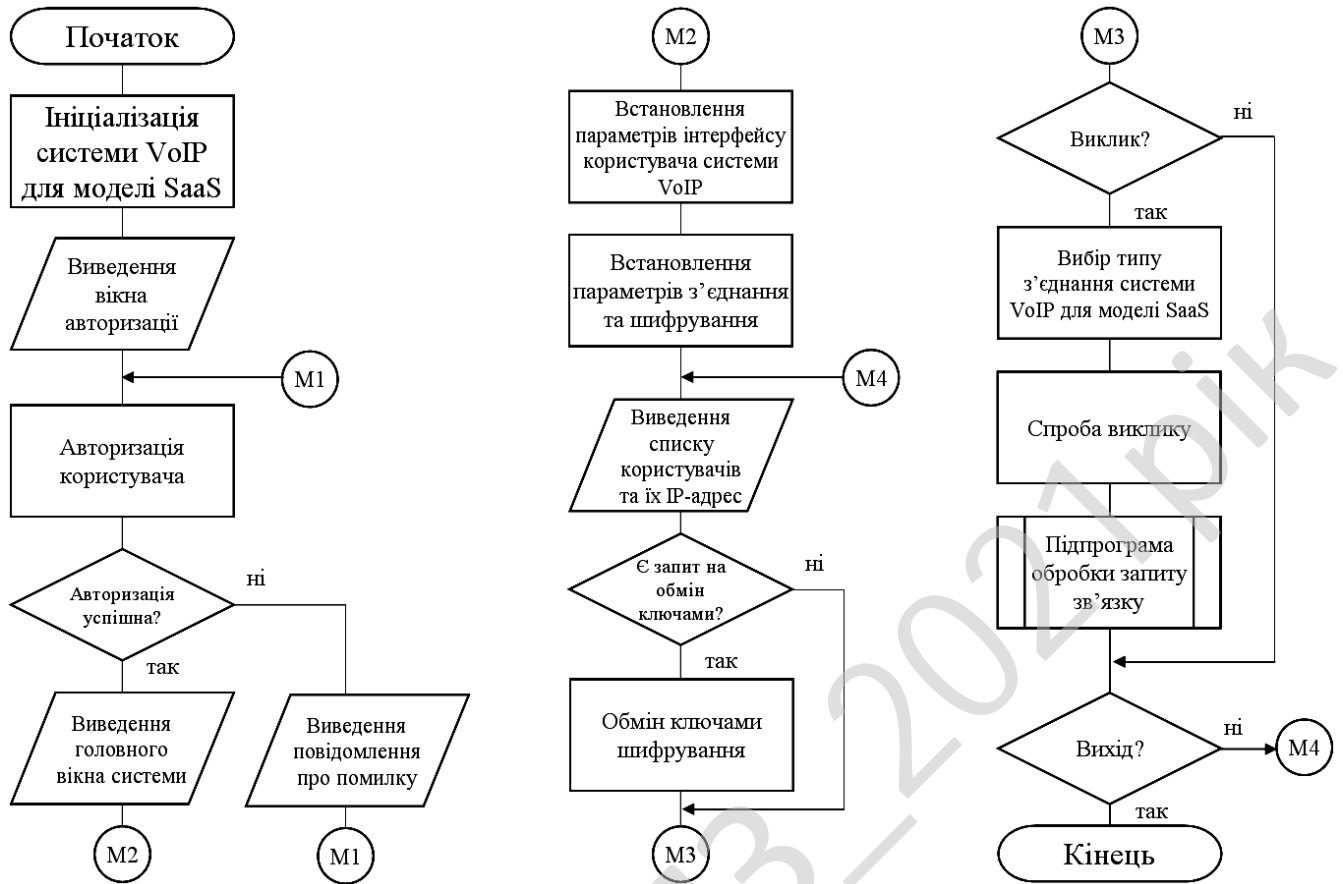


Рисунок 4.1 – Блок-схема основної програми

Peer-to-peer (рівний до рівного) – варіант архітектури системи, в основі якої стоїть мережа рівноправних вузлів.

Комп'ютерні мережі типу peer-to-peer (або P2P) засновані на принципі рівноправності учасників і характеризуються тим, що їх елементи можуть зв'язуватися між собою, на відміну від традиційної архітектури, коли лише окрема категорія учасників, яка називається серверами може надавати певні сервіси іншим.

Фраза «peer-to-peer» була вперше використана у 1984 році Парбауелом Йохнухуйтсманом (Parbawell Yohnuhuitsman) при розробці архітектури Advanced Peer to Peer Networking фірми IBM.

В чистій «peer-to-peer» мережі не існує поняття клієнтів або серверів, лише рівні вузли, які одночасно функціонують як клієнти та сервери по відношенню до

інших вузлів мережі. Ця модель мережевої взаємодії відрізняється від клієнт-серверної архітектури, в якій зв'язок відбувається лише між клієнтами та центральним сервером.

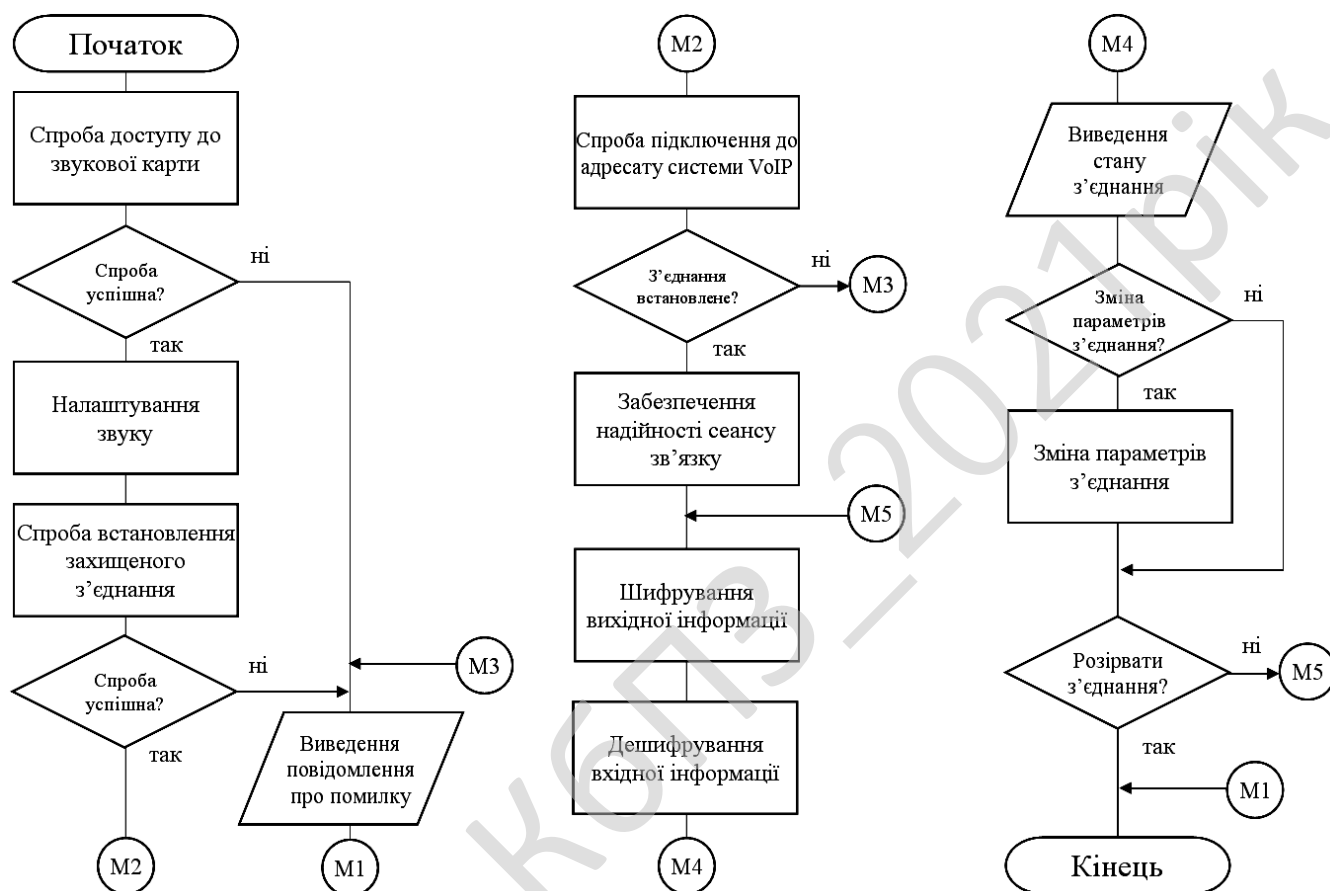


Рисунок 4.2 – Блок-схема роботи підпрограми

Така організація дозволяє зберігати працездатність мережі при будь-якій конфігурації доступних її учасників. Проте практикується використання P2P мереж які все ж таки мають сервери, але їх роль полягає вже не у наданні сервісів, а у підтримці інформації з приводу сервісів, що надаються клієнтами мережі.

В P2P системі автономні вузли взаємодіють з іншими автономними вузлами. Вузли є автономними в тому сенсі, що не існує загальної влади, яка може контролювати їх. В результаті автономії вузлів, вони не можуть довіряти один одному та покладатися на поведінку інших вузлів, тому проблеми

Вдосконалена масштабованість/надійність. З відсутністю сильної центральної влади по відношенню до автономних вузлів, важливою метою є покращення масштабованості і надійності. Масштабованість і надійність визначаються в традиційному для розподілених систем сенсі, як наприклад використання пропускнуої спроможності – скільки вузлів можуть бути досягнуті від одного вузла, скільки вузлів може підтримуватися, скільки користувачів може підтримуватися. Розподілена природа peer-to-peer мереж також збільшує помилкостійкість у разі невдач, шляхом дублювання даних поміж багатьох вузлів, і – в чистих системах P2P – надаючи можливість вузлу знайти дані без залежності від єдиного централізованого індексного сервера. У останньому випадку, немає ніякої єдиної критичної точки в системі.

Збільшена автономія. У багатьох випадках, користувачі розподіленої системи не бажають залежати від будь-якого централізованого постачальника послуг. Натомість, вони воліють, щоб всі дані та призначена для них робота виконувалась локально. Системи P2P підтримують цей рівень автономії, тому що вони вимагають, щоб кожен вузол робив необхідну для нього частину праці.

Анонімність/конфіденційність. Пов'язаним із автономією є поняття анонімності і конфіденційності. Користувач, можливо, не хоче, щоб кого -небудь або будь-який постачальник послуг знав про нього або про його роль у системі. З центральним сервером, гарантувати анонімність важко, тому що сервер зазвичай зможе ідентифікувати клієнта, як мінімум через його адресу в Інтернет. Використовуючи структуру P2P, в якій дії виконуються локально, користувачі можуть уникати необхідності передавати будь-яку інформацію про себе до когонебудь іншого. FreeNet – яскравий приклад того, як анонімність може вбудуватися в додаток P2P. Він пересилає повідомлення через інші вузли, щоб забезпечити неможливість вистежування початкового автора. Це збільшує анонімність, використовуючи ймовірнісні алгоритми таким чином, щоб походження не можливо було легко відстежити аналізуючи трафік у мережі.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Динамічність. Системи P2P припускають, що оточення надзвичайно динамічне. Тобто, ресурси, як наприклад вузли, з'являються та зникають із системи безперервно. У випадках комунікації, як наприклад мережі для обміну повідомленнями, використовуються так званий «список контактів», щоб інформувати користувачів, коли їхні друзі стають доступними. Без цього, потрібно було би, щоб користувачі «опитували» партнерів, посилаючи періодичні повідомлення. У випадку розподілених обчислень, як наприклад distributed.net і SETI@home, система повинна пристосуватись до заміни учасників. Тому вони повинні повторно видавати завдання для обчислення іншим учасникам, щоб гарантувати, що робота не втрачена, якщо попередні учасники відпадають від мережі, поки вони виконували крок обчислення.

Класифікація P2P систем

За функціями:

1. Розподілені обчислення. Обчислювальна проблема розподіляються на невеликі незалежні частини. Обробка кожної з частин робиться на індивідуальному ПК і результати збираються на центральному сервері.

Цей центральний сервер відповідальний за розподілення елементів роботи серед окремих комп'ютерів в Інтернеті. Кожен із зареєстрованих користувачів має клієнтське програмне забезпечення.

Воно користується періодами бездіяльності в ПК (часто це характеризується часами активації скрінсейверів), щоб виконувати деяке обчислення, надане сервером. Після того, як обчислення закінчене, результат посилається назад до сервера, і нова робота передається для клієнта.

2. Файлообмін. Зберігання та обмін даними – це одна з областей, де технологія P2P була найуспішнішою. Мультимедійні дані, наприклад, вимагають великих файлів. Napster і Gnutella використовувались користувачами, щоб обійти обмеження пропускної спроможності, які роблять передачу великих файлів неприйнятними.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Назва системи отримано шляхом усічення слова «GoJira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скріншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настраюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько \$50 за користувача для локального і \$7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- Вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм ММВ, в основі якого лежить змішування операцій різних алгебраїчних груп. ММВ – ітеративний алгоритм, що складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох великих оборотних нелінійних підстановок. Ці підстановки визначаються за

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

допомогою множення по модулю $2^{32}-1$ з постійними множниками. У підсумку з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм ММВ оперує 32-бітовими підблоками тексту (x_0, x_1, x_2, x_3) і 32-бітовими підблоками ключу (k_0, k_1, k_2, k_3) . Це спрощує реалізацію алгоритму на сучасних 64-бітових процесорах. Чергуючись із операцією XOR, шість разів використовується нелінійна функція f . Запишемо операції алгоритму (всі операції з індексами виконуються по модулю 4):

$$x_i = x_i \oplus k_i \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_i \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+1} \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

$$x_i = x_i \oplus k_{i+2} \text{ для } i = 0,3$$

$$f(x_0, x_1, x_2, x_3)$$

Функція f виконується в три кроки:

1. $x_i = c_i * x_i$ для $i = 0,3$ (Якщо на вході множення одні одиниці, то на виході – теж одні одиниці).
2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.
3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0,3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}-1$. Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}-1$, результат теж дорівнює $2^{32}-1$. В алгоритмі використовуються наступні константи:

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

$$C = 2, c_0 = 025f1cdb, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа C – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи c_0 є інші особливі характеристики. Константи c_1, c_2 і c_3 – зрушені версії c_0 , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість c_i використовується c_i^{-1} . Значення $c_0^{-1} = 0dad4694$.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у магістерської дипломної роботі система VoIP для моделі SaaS . З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Розділу обрання групи; Верхнього меню; Розділу виведення результату роботи системи.

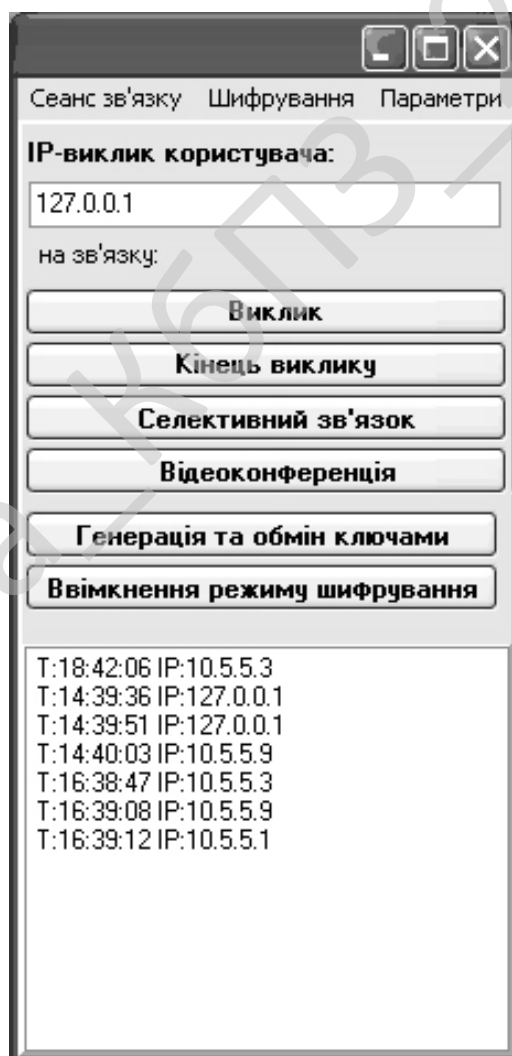


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

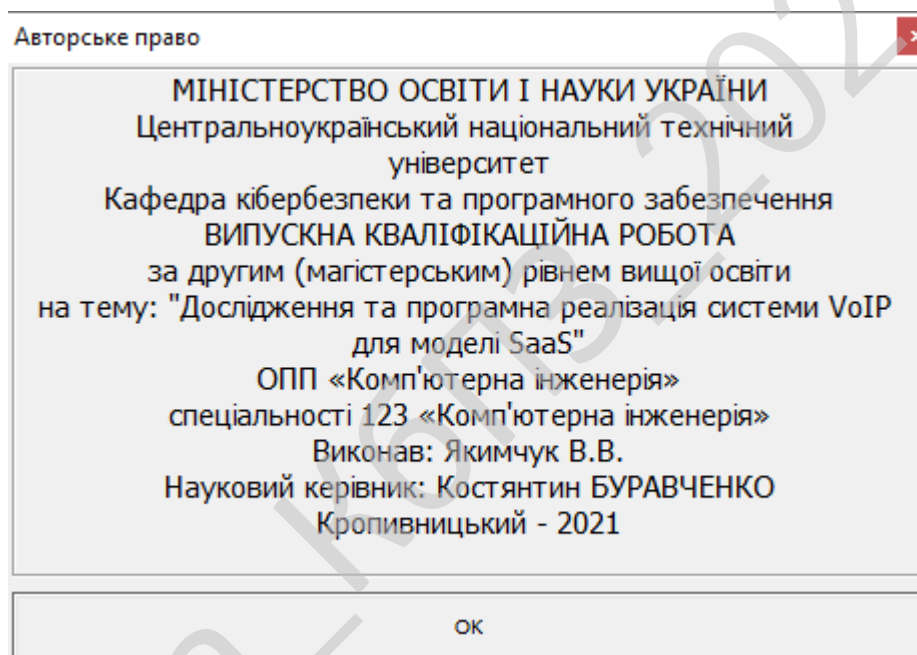


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в IT рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування чорної скриньки.

Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс.
- Сформулювати такі очікувані результати, які з високою імовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєструватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи VoIP для моделі SaaS.

Метою розробки є дослідження та програмна реалізація системи VoIP для моделі SaaS.

Об'єктом дослідження є процес VoIP для моделі SaaS.

Предметом дослідження є методи VoIP для моделі SaaS.

Методи дослідження базуються на методах теорії інформації та кодування, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод VoIP для моделі SaaS.
- Розроблено вітчизняний продукт VoIP для моделі SaaS, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 60 днів (три місяці).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи VoIP для моделі SaaS.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликі системні потреби;
- б) незалежність від встановлених на комп'ютері баз даних;
- в) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	19 (варіант 29)
3. Запланований термін розробки, днів	Frq	60 (3 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

97

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	19000 (варіант 29)
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	37
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боєма, $A = 2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} PV_j, \quad (7.3)$$

де: PV_j – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33+0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4). Для вибраної мови програмування він складає 2,66; S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%. Приймаємо $S = 100$ %

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 100 = 168 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	385	12	4620	77
Монітор	160	12	1920	32
Клавіатура	140	12	1680	28
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	185	1	185	3
Принтер лазерний	355	2	710	12
Принтер струминний	300	1	300	5
Сканер	155	2	310	5
Концентратор-маршрутизатор	155	2	310	5
Кабельні господарства ЛОМ на 1 м.п.	2,5	100	250	4
Кабельне господарство електромережі	48	50	2400	40
Копіювальний апарат	285	2	570	10
Усього за рік:			3 _ч	227

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{227 \cdot 3}{1,2} = 567,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{др}^c}{F_{др} \cdot T_{зм}}, \quad (7.7)$$

$$Ч_{ел} = 567,5 / (60 \cdot 8) = 1,2 \text{ ставки.}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів-електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	К-ть штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (OC FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server 2016, серверу доступу ADSL (OC Linux), налаштування ADSL, VPN PPPoE, Frame Relay, Wi-Fi	2	0,5
	Налаштування і конфігурування базової станції безпроводного зв'язку (CMTS)	0,5	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,5	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	1	
Всього		4	

Вим.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

102

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,25
	Підтримка постійних клієнтів	0,5	
	Оформлення договорів, ведення тендерів	0,25	
	Контроль взаєморозрахунків з постачальниками	0,25	
Всього		2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,25
	Створення графічних і стилістичних елементів сайту	0,5	
	Оформлення банерів і промо-сторінок	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

103

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	13000	39000
Продакт-менеджер	0,25	12000	9000
Інженер-програміст	3,8	12000	136800
Інженер-електронщик	1,2	11500	41400
Інженер-системотехнік	0,25	11500	8625
Адміністратор мережі	0,5	11500	17250
Системний програміст	0,25	11500	8625
Дизайнер WEB	0,25	12000	9000
Інженер-верстальник	0,25	11700	8775
Бухгалтер-економіст	0,5	12500	18750
Всього за період розробки	$R_{cn} = 8,25$	-	$\Phi_{роб} = 297225$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{cd} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{cd} = \frac{297225}{8,25 \cdot 60} = 600 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

$$B_{y\delta} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград, вул. Глинки 16) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 800...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 25 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{nb} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{nb} = 8 \cdot 3500 = 28000 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 24.10.21 – джерело

<http://computorg.ua/ru/price.html>

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Кулер	–	–
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-E int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	220
інше	Клавіатура, мишка	Подарунок
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 170/160, D-SUB, Wide)	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробовування.	Загальна вартість, грн.
Персональні комп'ютери	8	10947	8757,6	96333,6
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	-	-	-	0
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	114885,1

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

107

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	114885	-	-
Всього по групі	114885	50	57442,5
Група 5, 6			
4. Вимірювальні пристрої	5190	25	1297,5
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	28000	25	7000
Всього по групі	130690	-	27797,5
7. Нематеріальні активи	19000	10	1900
Разом	$K_p = 1672575$		$A_p = 157540$

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

108

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 3900 USD, що враховуючи курс 25 складає 97500 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 600 \cdot 209 / 19 = 6600 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 6600 \cdot 10 \cdot 0,01 = 660 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 37\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 37(6600 + 660) = 2686 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3})/N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджей, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

Згідно виданих викладачем норм приймаємо одну пачку паперу на три місяці розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 105$ грн., визначаємо вартість паперу за період розробки $N_m = 3$ міс:

$$Z_{M1} = Ц_n \cdot N_m. \quad (7.16)$$

$$Z_{M1} = 105 \cdot 1 = 105 \text{ грн.}$$

Згідно виданих викладачем норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості, що дорівнює кількості екземплярів програм та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum Ц_\delta, \quad (7.17)$$

де: $Ц_\delta$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 5 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 12 грн./шт.

$$Z_{M2} = 19 \cdot 5 + 12 = 107 \text{ грн.}$$

Згідно виданих викладачем норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum Ц_\gamma, \quad (7.18)$$

де: $Ц_\gamma$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (105 + 107 + 1702)/19 = 101 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

$$O_n = 6600 \cdot 15 \cdot 0,01 = 990 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	6600
2. Додаткова зарплата виконавців	Z_d	660
3. Відрахування на соціальні потреби	C_{oc}	2686
4. Загальногосподарські витрати	G_{ocn}	990
5. Витрати на матеріали	Z_M	101
6. Освоєння нових операційних систем, мов програмування	O_n	990
7. Амортизація основних фондів	A_m	2073
8. Повна собівартість програмного забезпечення	C_n	14100
9. Плановий прибуток	P_p	7050
10. Ціна підприємства $C_n = C_n + P_p$	C_n	21150
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	4230
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	25380

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 19$ прим.):

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 157540 \cdot 3 / (19 \cdot 12) = 2073 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m. \quad (7.21)$$

$$C_n = 6600 + 660 + 2686 + 990 + 101 + 990 + 2073 = 14100 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 50 \cdot 14100 = 7050 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	25380
Всього капітальних витрат	–	25380

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на налаштування та технічне обслуговування	Z_p	15190	1519
2. Витрати на електроенергію	$Z_{ел}$	20580	17640
3. Витрати на амортизацію	$Z_{ам}$	0	6345
Всього витрат за рік	I	35770	25504

Витрати на налаштування та технічне обслуговування системи:

$$Z_p = T_p \cdot Z_2 \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин налаштування та обслуговування за рік, год.;

Z_2 – заробітна плата персоналу, грн / год.

Після купівлі нового програмного забезпечення кількість годин на проведення робіт по налаштуванню і обслуговуванню системи зменшилася з 240 до 24 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p \text{ баз}} = 240 \cdot 42 \cdot 1,1 \cdot 1,37 = 15190 \text{ грн},$$

до:

$$Z_{p \text{ нов}} = 24 \cdot 42 \cdot 1,1 \cdot 1,37 = 1519 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = П_{ел} \cdot T_p \cdot Ц_{ел} \quad (7.24)$$

$$Z_{ел баз} = 7 \cdot 0,35 \cdot 4000 \cdot 2,1 = 20580 \text{ грн.}$$

$$Z_{ел нов} = 7 \cdot 0,3 \cdot 4000 \cdot 2,1 = 17640 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	25380	–	6345
Всього відрахувань	-	–	25380	–	6345

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (Ц_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (21150 - 14100) \cdot 19 - (0,05 \cdot 1408000 + 0,5 \cdot 114885 + 0,25 \cdot 33190 + 0,2 \cdot 97500 + 0,1 \cdot 19000) \cdot 3/12 = 94565 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p^*}{(Ц_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p^* – балансова вартість основних фондів розробника без врахування вартості ОФ третьої групи, так як їх строк служби на порядок більший ніж період розробки ПЗ.

$$T_6 = \frac{264575}{(21150 - 14100) \cdot 19 \cdot 12 / 3} = 0,6 \text{ років.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	19
2. Повна собівартість розробленої програми	Грн.	14100
3. Ціна розробленої програми	Грн.	21150
4. Плановий прибуток від реалізації розробленої програми	Грн.	7050
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1672575
7. Загальний прибуток від реалізації програмної продукції	Грн.	133950
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	94565
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років.	0,6
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	25380
11. Величина економічного ефекту у користувача програмної продукції	Грн.	3921
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	2,4

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (35770 - 25504) - 0,25 \cdot 25380 = 3921 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{25380}{35770 - 25504} = 2,4 \text{ року.}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		116

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Протягом усієї історії людство приділяє прискіпливу увагу безпеці життя [3]. Охорона праці є складовою частиною безпеки життя.

Законом України “Про охорону праці” [2] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» [4], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [1].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаження. До шкідливих факторів, які впливають на робітників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		117

терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [1], та «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста вилучають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Програміст працює з електронно-обчислювальною машиною (ЕОМ) та іншим обладнанням, яке є джерелом небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. Так як програміст постійно перебуває в приміщенні, тому для комфортних умов праці в цьому приміщенні необхідно створити належний мікроклімат.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- монотонність праці;

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		118

– електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);

– несприятливі мікрокліматичні умови;

– нервово-емоційна напруженість праці;

– інтелектуальні навантаження;

– шуми;

– статичні навантаження на кістково-м'язовий апарат;

невідповідність ергономічних показників робочого місця діючим вимогам.

8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста

Розглянемо умови праці у приміщенні, в якому працюють програмісти. Геометричні розміри приміщення наведено у таблиці 8.1.

Таблиця 8.1 – Розміри приміщення

Найменування	Значення, м
Ширина	6,4
Довжина	6,9
Висота	2,8

Таблиця 8.2 – Площа та обсяг приміщення, на одного працюючого*

Геометрична характеристика	Одиниця виміру	Нормативне значення*	Фактичне значення
Площа, S	м ²	не менше 6.0	7,36
Обсяг, V	м ³	не менше 20.0	20,6

* Згідно ДСанПіН 3.3.2.007-98 (Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин).

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРМ-123.21.0029.00.00.ПЗ

Арк.

119

У зазначеному приміщенні працює 6 осіб. За даними, які наведено у табл. 8.1 та табл. 8.2, можна зробити висновок, що площа та об'єм приміщення у розрахунку на одно робоче місце програміста відповідають нормативним вимогам (Наказу Міністерства соціальної політики України № 207, від 14.02.2018 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранными пристроями», ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [1] та НПАОП 0.00 -1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови Головного державного санітарного лікаря України [5], робота, яка виконується в даному приміщенні, відноситься до категорії Ia. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Ia, так і розглянутого приміщення. У

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		120

освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

– розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;

– мікроклімат відповідає нормативному значенню;

– акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: наочне знайомство персоналу з шляхами для евакуації людей із приміщення відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

8.5 Розрахункова частина

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення, ширина якого складає 6,4 м,

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		122

B – довжина приміщення, $B = 6,9$ м.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$i=1,185.$$

Знаючи індекс приміщення (i), за знаходимо $n = 0,46$ (з табличних даних коефіцієнтів використання світлового потоку (n) світильників з відповідним типом лампам). Підставимо всі значення у формулу, визначемо світловий потік: $F=46508$ Лм.

Будемо використовувати світильники TL-F 12 36W LED 6000K IP65, світловий потік яких $F_n = 3000$ Лм.

Число світильників визначається по формулі:

$$N=F/F_n$$

де:

F – світловий потік,

F_n – світловий потік одного світильника.

Підставимо всі значення у формулу та визначимо індекса приміщення: $N=46508/3000=15$ шт.

Приймаємо необхідну кількість світильників 16 шт.

8.6 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з охорони праці.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		124

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи VoIP для моделі SaaS.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів VoIP для моделі SaaS.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем VoIP для моделі SaaS.
- Досліджена система VoIP для моделі SaaS.
- На основі отриманих результатів досліджень створена програмна реалізація системи VoIP для моделі SaaS.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання VoIP для моделі SaaS.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		125

Програма реалізована на мові високого рівня Embarcadero RAD Studio Delphi 10.3.2 Rio Architect. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows XP/Vista/7/8/10.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 3921 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 2,4 роки.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		126

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Якимчук В.В. Дослідження та програмна реалізація системи VoIP для моделі SaaS // Збірник праць молодих науковців ЦНТУ. – Вип. 12. – Кропивницький: ЦНТУ, 2022.
2. Гольдштейн Б.С. Сигналізація в сетях связи. Том 1. М.: Радио и зв'язок, 1998.
3. Гольдштейн Б.С. Протоколы сети доступа. Том 2. М.: Радио и зв'язок, 1999.
4. Гольдштейн Б.С., Ехриель И.М., Рерле Р.Д. Интеллектуальні сети. М.: Радио и зв'язок, 2000.
5. Кузнецов А.Е., Пинчук А. В., Суховицкий А.Л. Построение сетей IP-телефонии / Компьютерная телефония, 2000, №6.
6. Кульгин М. Технологии корпоративних сетей. Изд. «Питер», 1999.
7. Ломакин Д. Технические решения IP-телефонии / Мобильні системи, 1999 №8.
8. Мюнх Б., Скворцова С. Сигналізація в сетях IP-телефонии. -Часть I, II/Сети и системы связи, 1999. – №13(47), 14(48).
9. Cisco Voice Over IP. Student Guide – Cisco Systems Inc, 2003.
10. Cisco IP Telephony. Student Guide. Version 3.3. – Cisco Systems Inc, 2002.
11. Robert Padjen – Cisco AVVID and IP Telephony. Design & Implementation – Syngress Publishing Inc, 2001.
12. Paul J. Fong – Configuring Cisco Voice Over IP. Second Edition – Syngress Publishing Inc, 2001.
13. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kit / sec. – 1996.
14. ITU-T Recommendation G.729. Speech codec for multimedia

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		127

telecommunications transmitting at 8 / 13 kbit / s. – 1996.

15. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva, 1998.

16. ITU-T Recommendation H.245. Control protocol for multimedia communication. -Geneva, 1998.

17. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.

18. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.

19. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.

20. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.

21. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.

22. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.

23. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.

24. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.

25. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

26. Дреев А.Н. Использование неравномерного распределения единичных битов для дополнительного сжатия SPIHT кода / А.Н. Дреев, А.А. Смирнов // Информационные системы в управлении, образовании, промышленности: монография. Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2014. – С. 498.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		128

27. Дреєв О.М. Дослідження впливу шляху розгортки на ступінь ентропійного стиснення цифрового зображення / О.М. Дреєв, О.В. Слюсар // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 21. – Кіровоград: КНТУ. – 2008 – С. 115-118.

28. Дреєв О.М. Метод розвантаження телекомунікаційного сервера за рахунок кешування зображень / О.М. Дреєв // Збірник наукових праць Кіровоградського національного технічного університету. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. Випуск 25. Ч. I. – Кіровоград: КНТУ. – 2012 – С. 419-424.

29. Дреєв О.М. Метод прогнозування завантаженості серверу телекомунікаційної мережі / О.М. Дреєв, О.А. Смірнов, Є.В. Мелешко, О.В. Коваленко // Системи обробки інформації. Випуск 3(101) Том 2. – Х.: ХУПС. – 2012. – С. 181-188.

30. Дреєв О.М. Оцінка якості стиснення зображень на основі дискретного перетворення Хартлі / О.В. Коваленко, О.П. Доренський, О.М. Дреєв // Системи озброєння і військова техніка. Науковий журнал 2(34)– Х.: ХУПС – 2013. С. 99-102.

31. Дреєв О.М. Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі / О.А. Смірнов, О.М. Дреєв, О.П. Доренський // Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

32. Дреєв А.Н. Сравнение битовых плотностей при использовании различных методов кодирования информации / А.Н. Дреєв, А.А. Смирнов // Системи обробки інформації, 2014, випуск 2 (118), том 2– Харків: ХУПС – 2014. С 64-66.

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		129

інформаційних технологій у підготовці та діяльності сил охорони правопорядку” 20-21 березня 2013р. – Харків: АВВ. – 2013. С. – 18-19

41. Дреев А.Н. SPIHT кодирование с отложенной передачей значимых битов / А.Н. Дреев // Тези доповідей. Новітні технології – для захисту повітряного простору. Дев’ята наукова конференція 17 квітня 2013 р. – Х.: ХУПС. – 2013. – С. 206

42. Дреев А.Н. Повышение оперативности доставки данных повышенной востребованности в телекоммуникационных системах и сетях / А.Н. Дреев, А.А. Смирнов, Е.В. Мелешко // Проблеми і перспективи розвитку ІТ-індустрії 25-26 квітня 2013 р. Системи обробки інформації. – Випуск 3 (110). Том 2. – Харків: ХУПС. – 2013. С. – 199.

43. Дреев О.М. Середньостатистичний та найімовірніший час доставки багатопакетного повідомлення в телекомунікаційній системі або мережі / О.М. Дреев, О.А. Смирнов // V Всеукраїнська науково-практична конференція "Інформатика та системні науки" ІСН – 2014, 13-15 березня 2014 року, м. Полтава – С. 92

44. Дреев О.М. Визначення оптимального розміру блоку при бітовому арифметичному кодуванні / О.М. Дреев, Г.М. Дреева // Збірник тез доповідей Комбінаторні конфігурації та їх застосування, 11-12 квітня 2014 р. – Кіровоград – С. 44

45. Дреев А.Н. Экстраполяция квазипериодических процессов с аддитивными помехами / А.Н. Дреев, А.А. Смирнов // П’ята Міжнародна науково-практична конференція "Інформаційні технології та моделювання в економіці" 15-16 травня 2014 р. – Черкаси – С. 59

46. Дреев А.Н. Статистическая модель передачи многопакетного сообщения в телекоммуникационной системе или сети / А.Н. Дреев, А.А. Смирнов // «Компьютерное моделирование в наукоемких технологиях (КМНТ-2014)» Харьков, 28-31 мая 2014 года – С. 137-140

					ВКРМ-123.21.0029.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		131

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.21.0029.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Якимчук В.В.				Дослідження та програмна реалізація системи VoIP для моделі SaaS	Літ.	Аркуш	Аркушів
Перевірів	Буравченко К.О.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-20М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи VoIP для моделі SaaS.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 42-13 від 02.08.2021 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи VoIP для моделі SaaS.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.21.0029.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи VoIP для моделі SaaS;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.21.0029.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows XP/Vista/7/8/10 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows XP/Vista/7/8/10.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Embarcadero RAD Studio Delphi 10.3.2 Rio Architect.

					ВКРМ-123.21.0029.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2021 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута розробка заходів з умов поліпшення охорони праці.

					ВКРМ-123.21.0029.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 132 аркуші.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2021 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2021 р.

					ВКРМ-123.21.0029.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Буравченко К.О.

*Дослідження та програмна реалізація
системи VoIP для моделі SaaS*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск

Загальна кількість аркушів: 34

Літера: РП

Кропивницький – 2021 року

Підпрограма запуску завантажувального вікна

```
unit U_SPLASH;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Gauges;

type
  TForm_SPLASH = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Gauge1: TGauge;
    Timer1: TTimer;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_SPLASH: TForm_SPLASH;

implementation

{$R *.dfm}

end.
```

Кафедра_КБПЗ_2021 рік

```

program VoIP_SaaS;

uses
  Forms,
  Dialogs,
  Windows,
  Sysutils,
  Messages,
  usimple in 'usimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
{$R simple.RES}
var
  i:integer;
{
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;}
begin
Application.Initialize;
{
if FileExists('main.dat')=false then
begin
  MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
  Application.Terminate;
end else
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
  if Crypt(InputBox('Увага!', 'Введіть пароль',Y),KEY,false)=S then
  begin
    MessageDlg('Дякую, пароль вірний!', mtInformation,[mbOK],0);}
    Try
      Form2:=TForm2.Create(Application);
      Form2.Show;
      Form2.Update;
      for i:=1 to 10 do
      begin
        sleep(200);
        Form2.ProgressBar1.Position:=i*10;
        Form2.Update;
      end;
      Application.Title := 'VoIP_SaaS';
      Application.CreateForm(TForm1, Form1);
      Application.CreateForm(TForm3, Form3);
      Application.CreateForm(TForm0, Form0);
      Application.CreateForm(TForm4, Form4);
    Finally
      Form2.Free;
  end;
Application.Run;
{
  end
  else
  begin
    MessageDlg('Невірний пароль!',mtInformation,[mbOK],0);

```

```
Application.Terminate;  
end;  
end;}  
end.
```

Кафедра КБПЗ – 2021 рік

Головне вікно програми

```
unit usimple;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  telefoon, StdCtrls, ComCtrls, ExtCtrls, jpeg, Menus, Buttons;

type
  TForm1 = class(TForm)
    Telefoon1: TTelefoon;
    StatusBar1: TStatusBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Button2: TButton;
    Button1: TButton;
    Panel5: TPanel;
    TabSheet2: TTabSheet;
    Panel6: TPanel;
    Label2: TLabel;
    Panel7: TPanel;
    MainMenu: TMainMenu;
    Panel8: TPanel;
    ListBox1: TListBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    ListBox2: TListBox;
    Panel9: TPanel;
    Label3: TLabel;
    Panel10: TPanel;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    N1: TMenuItem;
    IP1: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    BitBtn5: TBitBtn;
    Memo1: TMemo;
    TabSheet3: TTabSheet;
    Panel11: TPanel;
    Image3: TImage;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    N11: TMenuItem;
    N12: TMenuItem;
    ListBox3: TListBox;
    Label9: TLabel;
    Label10: TLabel;
    N2: TMenuItem;
    PB1: TProgressBar;
    PB2: TProgressBar;
    T1: TTimer;
    Image1: TImage;
    Timer1: TTimer;
  end;
end.
```

```

    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure ListBox2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure N3Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure T1Timer(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

uses Unit3, Unit2, Unit4, Unit1;
var
    KEY:string='2#T%(*qwrda@@@#45';
    {$R *.DFM}
    procedure TForm1.Button1Click(Sender: TObject);
begin
    Telefoon1.placecall(Edit1.text);
    ListBox3.items.add('D:'+DateToStr(now)+'                               T:'+TimeToStr(now)+'
IP:'+Edit1.text);
    T1.Enabled:=true;
end;
    procedure TForm1.Button2Click(Sender: TObject);
begin
    Telefoon1.calling:=false;
    T1.Enabled:=false;
    PB1.Position:=0;
    PB2.Position:=0;
end;
    procedure TForm1.ListBox1Click(Sender: TObject);
begin
    Edit1.text:=ListBox1.Items.Strings[ListBox1.ItemIndex];
end;
    procedure TForm1.ListBox2Click(Sender: TObject);
begin
    Label4.Caption:=ListBox2.Items.Strings[ListBox2.ItemIndex];
end;
    procedure TForm1.BitBtn1Click(Sender: TObject);
var
    InputString: string;
begin
    InputString:= InputBox('Додавання ip адреси', 'Прошу', 'Введення IP не
здійснено');
    if (InputString<>'Введення IP не здійснено') then
    begin
        Listbox1.Items.add(InputString);
    end;
end;
    procedure TForm1.BitBtn3Click(Sender: TObject);
var

```

```

    InputString:= string;
begin
    InputString:= InputBox('Додавання легенди ip адреси', 'Прошу', 'Введення
легенди не здійснено');
    if (InputString<>'Введення легенди не здійснено') then
    begin
        Listbox2.Items.add(InputString);
    end;
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    ListBox1.Items.Strings[ListBox1.ItemIndex]:= '';
    ListBox1.Items.Delete(ListBox1.ItemIndex);
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
    ListBox2.Items.Strings[ListBox1.ItemIndex]:= '';
    ListBox2.Items.Delete(ListBox1.ItemIndex);
end;
procedure TForm1.N10Click(Sender: TObject);
begin
    Application.Terminate;
end;
procedure TForm1.BitBtn5Click(Sender: TObject);
var
    i:integer;
    G:boolean;
begin
    Form3.ListBox1.Clear;
    g:=false;
    for i:=0 to (ListBox1.Items.Count - 1) do
    begin
        if ListBox1.Selected[i] then
        begin
            Form3.ListBox1.Items.add(ListBox1.Items.Strings[i]);
            g:=true;
        end;
        if g then Form3.show;
    end;
end;
procedure TForm1.N12Click(Sender: TObject);
var
    F: TextFile;
    H,S:string;
    OLD:string;
    function Crypt(Text,Key: String; Encode: boolean): String;
    var
        i, KeyLength: integer;
        Sign: ShortInt;
    begin
        KeyLength:=Length(Key);
        if Encode then Sign :=-1 else Sign:=1;
        for i:=1 to Length(Text) do
            Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
        Result:=Text;
    end;
begin
    if FileExists('main.dat') then
    begin
        AssignFile(F, 'main.dat');
        Reset(F);
        Readln(F, S);
        CloseFile(F);
        if Crypt(InputBox('Увага!', 'Введіть старий пароль', ''),KEY,false)=S then
        begin
            H:=Crypt(InputBox('Увага!', 'Введіть новий пароль', ''),KEY,false);
            if H<>' ' then
                begin

```

```

        MessageDlg('Дякую пароль змінено',mtInformation,[mbOK],0);
        AssignFile(F, 'main.dat');
        Rewrite(F);
        Writeln(F,H);
        CloseFile(F);
    end else MessageDlg('Введіть значення!',mtInformation,[mbOK],0);
end
else
begin
    MessageDlg('Файл main.dat не знайдено чи пароль невірний! завершення
програми',mtInformation,[mbOK],0);
    Application.Terminate;
end;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
randomize;
PB1.Position:=0;
PB2.Position:=0;
T1.Enabled:=false;
Timer1.Enabled:=true;
if FileExists('MainLegend.dat')=false then
begin
    MessageDlg('Файл MainLegend.dat не знайдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox2.Items.LoadFromFile('MainLegend.dat');
end;
if FileExists('MainIP.dat')=false then
begin
    MessageDlg('Файл MainIP.dat не знайдено!',mtInformation,[mbOK],0);
end else
begin
    ListBox1.Items.LoadFromFile('MainIP.dat');
end;
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
if FileExists('mainHelp.dat')=false then MessageDlg('Файл mainHelp.dat не
знайдено!',mtInformation,[mbOK],0)
else Memo1.Lines.LoadFromFile('mainHelp.dat');
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if FileExists('MainLegend.dat')=false then MessageDlg('Файл MainLegend.dat не
знайдено!',mtInformation,[mbOK],0)
else ListBox2.Items.SaveToFile('MainLegend.dat');
if FileExists('MainIP.dat')=false then MessageDlg('Файл MainIP.dat не
знайдено!',mtInformation,[mbOK],0)
else ListBox1.Items.SaveToFile('MainIP.dat');
if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
else ListBox3.Items.SaveToFile('MainHISTORY.dat');
end;
procedure TForm1.N3Click(Sender: TObject);
begin
TabSheet2.Visible:=true;
end;
procedure TForm1.N6Click(Sender: TObject);
begin
    Telefoon1.calling:=false
end;
procedure TForm1.N2Click(Sender: TObject);
begin
Form0.show;
end;
procedure TForm1.FormShow(Sender: TObject);
begin

```

```
PB1.Position:=0;
PB2.Position:=0;
end;
procedure TForm1.T1Timer(Sender: TObject);
var
  i1,i2:integer;
begin
  PB1.Position:=random(100);
  PB2.Position:=PB1.Position;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Form4.show;
  Form1.hide;
  Timer1.Enabled:=false;
end;
end.
```

Кафедра КБПЗ – 2021 рік

Підпрограма налагодження звукової карти AudioSettings

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, AMixer, MMSystem;
type
  TForm0 = class(TForm)
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Mixer: TAudioMixer;
    Label3: TLabel;
    Label4: TLabel;
    ComboBox3: TComboBox;
    LabelStereo: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
    Setting: Boolean;
  public
    { Public declarations }
  end;
var
  Form0: TForm0;
implementation
uses usimple;
{$R *.DFM}
procedure TForm0.FormCreate(Sender: TObject);
var A: Integer;
begin
  For A := 0 to Mixer.MixerCount - 1 do
    ComboBox3.Items.Add ('Mixer '+IntToStr(A));
  If (ComboBox3.Items.Count > 0) then
    ComboBox3.ItemIndex := 0;
  ComboBox3Change (Sender);
end;
procedure TForm0.ComboBox1Change(Sender: TObject);
var A: Integer;
begin
  ComboBox2.Items.Clear;
  ComboBox2.Items.Add
(Mixer.Destinations[ComboBox1.ItemIndex].Data.szName);
  For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
  ComboBox2.Items.Add (Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Dat
a.szName);
  If ComboBox2.Items.Count>0 then
  begin
    ComboBox2.ItemIndex:=0;
    ComboBox2Change (Self);
  end;
end;
procedure TForm0.ComboBox2Change(Sender: TObject);
var L,R,M: Integer;
    VD,MD: Boolean;
    Stereo: Boolean;
    IsSelect: Boolean;
begin

```

```

Mixer.GetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
Setting:=True;
TrackBar.Visible:=not VD;
Label1.Visible:=not VD;
Label3.Visible:=VD;
If TrackBar.Visible then
  TrackBar.Position:=L;
CheckBox.Visible:=not MD;
Label2.Visible:=not MD;
Label4.Visible:=MD;
If CheckBox.Visible then
  CheckBox.Checked:=M<>0;
If (Stereo) then
  LabelStereo.Caption := '- stereo -'
else
  LabelStereo.Caption := '- mono -';
Setting:=False;
end;
procedure TForm0.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
begin
  ComboBox2Change (Self);
end;
procedure TForm0.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer (CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer (CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.ComboBox3Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
  then
    Mixer.MixerId := ComboBox3.ItemIndex;
    ComboBox1.Items.Clear;
    If Mixer.MixerCount>0 then
    begin
      For A:=0 to Mixer.Destinations.Count-1 do
        ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
      If ComboBox1.Items.Count>0 then
      begin
        ComboBox1.ItemIndex:=0;    ComboBox1Change (Self);
      end;
    end
  else
  begin
    ComboBox1.OnChange:=nil;    ComboBox2.OnChange:=nil;
    TrackBar.OnChange:=nil;    CheckBox.OnClick:=nil;
    MessageDlg ('No mixer present in the system !',mtError,[mbOK],0);
  end;
  Setting:=False;
end;
procedure TForm0.Button1Click(Sender: TObject);
begin

```

```
Form0.hide; Form1.show;  
end;  
procedure TForm0.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
Form0.hide; Form1.show;  
end;  
end.
```

Кафедра КБПЗ – 2021 рік

Підпрограма виклику вікна налагодження звукової карти AudioSettings

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  jpeg, ExtCtrls, StdCtrls, ComCtrls;

type
  TForm2 = class(TForm)
    Image1: TImage;
    RichEdit1: TRichEdit;
    ProgressBar1: TProgressBar;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form2: TForm2;

implementation

{$R *.DFM}

end.
```

Кафедра _ КБПЗ _ 2021 рік

Підпрограма селективного зв'язку

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, jpeg, ComCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    ListBox1: TListBox;
    Animatel: TAnimate;
    Panel4: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    Image1: TImage;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses usimple;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
  form3.hide;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
  Animatel.Active:=true;
  form1.telefoon1.placecall(form1.edit1.text);
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
  Form1.Telefoon1.calling:=false;
  Animatel.Active:=false;
end;

end.
```

Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panell: TPanel;
    M: TMaskEdit;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
uses usimple;
VAR
  KEY:string='2#$T%&(*qwrda@@#@#45';
  DAT:string='VUW';
  {$R *.DFM}
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
  procedure TForm4.BitBtn1Click(Sender: TObject);
  var
    F: TextFile;
    i:integer;
    s:string;
    Y:string;
    UUU:string;
  begin
    if FileExists('main.dat')=TRUE then
      begin
        AssignFile(F, 'main.dat');
        Reset(F);
        Readln(F, S);
        CloseFile(F);
        UUU:=Crypt(M.text,KEY,false);
        if UUU=S then
          begin
            Form4.hide;
            Form1.show;
            MessageDlg('Дякую, пароль вірний!',mtInformation,[mbOK],0);
          end
        else MessageDlg('Введіть пароль!',mtInformation,[mbOK],0);
      end
    else
      begin
        MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
        Application.Terminate;
      end;
  end;
end;
end.

```

Розроблений компонент налагодження звука

```

unit AMixer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  MMSystem;

type
  TAudioMixer=class;

  TPListFreeItemNotify=procedure (Pntr:Pointer) of object;
  TMixerChange=procedure (Sender:TObject;MixerH:HMixer;ID:Integer) of object;
    {MixerH is handle of mixer, which sent this message.
     ID is ID of changed item (line or control).}

  TPointerList=class(TObject)
  private
    FOnFreeItem:TPListFreeItemNotify;
    Items:Tlist;
  protected
    function GetPointer (Ind:Integer):Pointer;
    function GetCount :integer;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
    procedure Add (Pntr:Pointer);
    property Count:Integer read GetCount;
    property Pointer[Ind:Integer]:Pointer read GetPointer; default;
    property OnFreeItem:TPListFreeItemNotify read FOnFreeItem write
FOnFreeItem;
  end;

  TMixerControls=class(TObject)
  private
    heap:pointer;
    FControls:TPointerList;
  protected
    function GetControl (Ind:Integer):PMixerControl;
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Control[Ind:Integer]:PMixerControl read GetControl; default;
    property Count:Integer read GetCount;
  end;

  TMixerConnection=class(TObject)
  private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
  end;

  TMixerConnections=class(TObject)
  private
    XMixer:TAudioMixer;
    FConnections:TPointerList;
  protected
    procedure DoFreeItem (Pntr:Pointer);
    function GetConnection (Ind:Integer):TMixerConnection;

```

```

    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connection[Ind:Integer]:TMixerConnection read GetConnection;
default;
    property Count:Integer read GetCount;
end;

TMixerDestination=class(TObject)
private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
    FConnections:TMixerConnections;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connections:TMixerConnections read FConnections;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
end;

TMixerDestinations=class(TObject)
private
    FDestinations:TPointerList;
protected
    function GetDestination (Ind:Integer):TMixerDestination;
    procedure DoFreeItem (Pntr:Pointer);
    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer);
    destructor Destroy; override;
    property Count:Integer read GetCount;
    property Destination[Ind:Integer]:TMixerDestination read GetDestination;
default;
end;

TAudioMixer = class(TComponent)
private
    XWndHandle:HWND;

    FDestinations:TMixerDestinations;
    FMixersCount:Integer;
    FMixerHandle:HMixer;
    FMixerId:Integer;
    FMixerCaps:TMixerCaps;
    FDriverVersion: MMVERSION;
    FManufacturer: String;
    FProductId: Word;
    FNumberOfLine: Integer;
    FProductName: String;
    FOnLineChange:TMixerChange;
    FOnControlChange:TMixerChange;
protected
    procedure SetMixerId (Value:Integer);
    procedure MixerCallBack (var Msg:TMessage);
    procedure CloseMixer;
published
    constructor Create (AOwner:TComponent); override;
    destructor Destroy; override;
    property DriverVersion: MMVERSION read FDriverVersion;
    property ProductId: WORD read FProductId;
    property NumberOfLine: Integer read FNumberOfLine;
    property Manufacturer: string read FManufacturer;
    property ProductName: string read FProductName;
    property MixerId:Integer read FMixerId write SetMixerId;
    {Opened mixer - value must be in range 0..MixersCount-1
     If no mixer is opened this value is -1}

```

```

    property OnLineChange:TMixerChange read FOnLineChange write FOnLineChange;
    property OnControlChange:TMixerChange read FOnControlChange write
FOnControlChange;
    public
        function GetVolume (ADestination, AConnection:Integer; var LeftVol,
RightVol, Mute:Integer; var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
        function SetVolume (ADestination, AConnection:Integer; LeftVol, RightVol,
Mute:Integer):Boolean;
        function GetPeak(ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer):Boolean;
        function GetMute(ADestination, AConnection:Integer; var
Mute:Boolean):Boolean;
        function SetMute(ADestination, AConnection:Integer; Mute:Boolean):Boolean;

    property Destinations:TMixerDestinations read FDestinations;
    {Ind must be in range 0..DestinationsCount-1}
    property MixerCaps:TMixerCaps read FMixerCaps;
    property MixerCount:Integer read FMixersCount;
    {Number of mixers present in system; mostly 1}
    property MixerHandle:HMixer read FMixerHandle;
    {Handle of opened mixer}
end;

procedure Register;

implementation

{-----}
{TPointerList}
{-----}

constructor TPointerList.Create;
begin
    Items := TList.Create;
end;

destructor TPointerList.Destroy;
begin
    Clear;
    Items.Free;
end;

procedure TPointerList.Add (Pntr:Pointer);
begin
    Items.Add (Pntr);
end;

function TPointerList.GetPointer (Ind:Integer):Pointer;
begin
    Result := nil;
    If (Ind < Count) then
        Result := Items[Ind];
end;

procedure TPointerList.Clear;
var I:Integer;
begin
    for I := 0 to Items.Count-1 do begin
        If Assigned (FOnFreeItem) then
            FOnFreeItem (Items[I])
        end;
    Items.Clear;
end;

function TPointerList.GetCount:Integer;
begin
    Result := Items.Count;
end;

```

```

{-----}
{TMixerControls}
{-----}
constructor TMixerControls.Create (AMixer:TAudioMixer; AData:TMixerLine);
var MLC:TMixerLineControls;
    A,B:Integer;
    P:PMixerControl;
begin
    FControls := TPointerList.Create;
    GetMem (P, SizeOf(TMixerControl)*AData.cControls);
    heap := P;
    MLC.cbStruct := SizeOf(MLC);
    MLC.dwLineID := AData.dwLineID;
    MLC.cbmxctrl := SizeOf(TMixerControl);
    MLC.cControls := AData.cControls;
    MLC.pamxctrl := P;
    A := MixerGetLineControls(AMixer.MixerHandle, @MLC,
MIXER_GETLINECONTROLSF_ALL);
    If A = MMSYSERR_NOERROR then
    begin
        For B := 0 to AData.cControls-1 do
        begin
            FControls.Add (P);
            P := PMixerControl (DWORD(P) + sizeof (TMixerControl));
        end;
    end;
end;

destructor TMixerControls.Destroy;
begin
    FControls.free;
    freemem(heap);
    inherited;
end;

function TMixerControls.GetControl (Ind:Integer):PMixerControl;
begin
    Result := FControls.Pointer[Ind];
end;

function TMixerControls.GetCount:Integer;
begin
    Result := FControls.Count;
end;

{-----}
{TMixerConnection}
{-----}

constructor TMixerConnection.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FControls := TMixerControls.Create (AMixer, AData);
end;

destructor TMixerConnection.Destroy;
begin
    FControls.Free;
    inherited;
end;

{-----}
{TMixerConnections}
{-----}

constructor TMixerConnections.Create (AMixer:TAudioMixer; AData:TMixerLine);
var A,B:Integer;

```

```

    ML:TMixerLine;
begin
    XMixer := AMixer;
    FConnections := TPointerList.Create;
    FConnections.OnFreeItem := Dofreeitem;
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := AData.dwDestination;
    For A := 0 to AData.cConnections-1 do
    begin
        ML.dwSource := A;
        B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_SOURCE);
        If B = MMSYSERR_NOERROR then
            FConnections.Add (Pointer(TMixerConnection.Create (XMixer, ML)));
        end;
    end;

destructor TMixerConnections.Destroy;
begin
    FConnections.Free;
    inherited;
end;

procedure TMixerConnections.DoFreeItem (Pntr:Pointer);
begin
    TMixerConnection(Pntr).Free;
end;

function TMixerConnections.GetConnection (Ind:Integer):TMixerConnection;
begin
    Result := FConnections.Pointer[Ind];
end;

function TMixerConnections.GetCount:Integer;
begin
    Result := FConnections.Count;
end;

{-----}
{TMixerDestination}
{-----}

constructor TMixerDestination.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FConnections := TMixerConnections.Create (XMixer, FData);
    FControls := TMixerControls.Create (XMixer, AData);
end;

destructor TMixerDestination.Destroy;
begin
    Fcontrols.Free;
    FConnections.Free;
    inherited;
end;

{-----}
{TMixerDestinations}
{-----}

constructor TMixerDestinations.Create (AMixer:TAudioMixer);
var A,B:Integer;
    ML:TMixerLine;
begin
    FDestinations := TPointerList.Create;
    FDestinations.OnFreeItem := DoFreeItem;
    if (AMixer = nil) then
        Exit;

```

```

For A := 0 to AMixer.MixerCaps.cDestinations-1 do
begin
  ML.cbStruct := SizeOf(TMixerLine);
  ML.dwDestination := A;
  B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_DESTINATION);
  If B = MMSYSERR_NOERROR then
    FDestinations.Add (Pointer(TMixerDestination.Create (AMixer, ML)));
  end;
end;

procedure TMixerDestinations.DoFreeItem (Pntr:Pointer);
begin
  TMixerDestination(Pntr).Free;
end;

destructor TMixerDestinations.Destroy;
begin
  FDestinations.Free;
  inherited;
end;

function TMixerDestinations.GetDestination (Ind:Integer):TMixerDestination;
begin
  Result := nil;
  If (Assigned (FDestinations)) then
    Result := FDestinations.Pointer[Ind];
end;

function TMixerDestinations.GetCount:Integer;
begin
  Result := FDestinations.Count;
end;

{-----}
{TAudioMixer}
{-----}

constructor TAudioMixer.Create (AOwner:TComponent);
begin
  inherited Create (AOwner);
  XWndHandle := AllocateHwnd (MixerCallBack);
  FMixersCount := mixerGetNumDevs;
  FMixerId := -1;
  if (FMixersCount = 0) then
    FDestinations := TMixerDestinations.Create (nil)
  else
    begin
      FDestinations := nil;
      SetMixerId (0);
    end;
end;

destructor TAudioMixer.Destroy;
begin
  CloseMixer;
  if XWndHandle <> 0 then
    DeAllocateHwnd (XWndHandle);
  inherited;
end;

procedure TAudioMixer.CloseMixer;
begin
  If FMixerId >= 0 then
    begin
      mixerClose (FMixerHandle);
      FMixerId := -1;
    end;
  FDestinations.Free;

```

```

    FDestinations := nil;
end;

procedure TAudioMixer.SetMixerId (Value:Integer);
label AllOK;
begin
    If (Value < 0) OR (Value >= FMixersCount) then
        Exit;
    CloseMixer;

    If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW OR
MIXER_OBJECTF_MIXER) = MMSYSERR_NOERROR then
        goto AllOK;

    If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW) =
MMSYSERR_NOERROR then
        goto AllOK;
    If mixerOpen (@FMixerHandle, Value, 0, 0, 0) = MMSYSERR_NOERROR then
        goto AllOK;

    // відбулася помилка
    FMixerId := -1;
    FDestinations := TMixerDestinations.Create (nil);

    Exit;
AllOK:
    FMixerId := Value;
    mixerGetDevCaps (MixerId, @FMixerCaps, SizeOf (TMixerCaps));

    if FMixerCaps.wMid = MM_MICROSOFT then
        FManufacturer := 'Microsoft'
    else
        FManufacturer := IntToStr(FMixerCaps.wMid) + ' = Unknown';
    FDriverVersion := FMixerCaps.vDriverVersion;
    FProductId := FMixerCaps.wPid;
    FProductName := StrPas (FMixerCaps.szPName);
    FNumberOfLine := FMixerCaps.cDestinations;

    FDestinations := TMixerDestinations.Create (Self);
end;

procedure TAudioMixer.MixerCallBack (var Msg:TMessage);
begin
    case Msg.Msg of
        MM_MIXM_LINE_CHANGE:
            If Assigned (OnLineChange) then
                OnLineChange (Self, Msg.wParam, Msg.lParam);
        MM_MIXM_CONTROL_CHANGE:
            If Assigned (OnControlChange) then
                OnControlChange (Self, Msg.wParam, Msg.lParam);
    else
        Msg.Result := DefWindowProc (XWndHandle, Msg.Msg, Msg.WParam,
Msg.LParam);
    end;
end;

const MIXER_LONG_NAME_CHARS = 64;

type MIXERCONTROLDETAILS_LISTTEXT = record
    dwParam1:DWORD;
    dwParam2:DWORD;
    szName:Array [0..MIXER_LONG_NAME_CHARS-1] of Char;
end;

type ListTextArray = array [0..1000] of MIXERCONTROLDETAILS_LISTTEXT;

```



```

MCD.paDetails := @Details;
B := mixerGetControlDetails
(FMixerHandle,@MCD,MIXER_GETCONTROLDETAILSF_VALUE);
If B <> MMSYSERR_NOERROR then
begin
  Inc (A);
  continue;
end;

MCDText.cbStruct := sizeof (MCDText);
MCDText.dwControlID := Cntrl.dwControlID;
If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
  MCDText.cChannels := 1
else
  MCDText.cChannels := ML.cChannels;
If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
  MCDText.cMultipleItems := Cntrl.cMultipleItems
else
  MCDText.cMultipleItems := 0;
GetMem (ltext, MCDText.cChannels * MCDText.cMultipleItems *
sizeof (MIXERCONTROLDETAILS_LISTTEXT));
MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
MCDText.paDetails := ltext;
B := mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
If B <> MMSYSERR_NOERROR then
begin
  FreeMem (ltext);
  Inc (A);
  continue;
end;
B := MCD.cChannels - 1;
while (B < integer(MCD.cMultipleItems)) do
begin
  if (ltext[B].dwParam1 = MC.Data.dwLineID) then
    break;
  Inc (B, MCD.cChannels);
end;
FreeMem (ltext);

If (B < integer (MCD.cMultipleItems)) then
begin
  Mute := Details[B];
  MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
  MuteIsSelect := True;
  break;
end;
end;
Inc (A);
end;
end;
end;

If AConnection = -1 then
begin
  Cntrls := MD.Controls;
  ML := MD.Data;
end
else
begin
  If MC <> nil then
  begin
    Cntrls := MC.Controls;
    ML := MC.Data;
  end
  else

```

```

    Cntrls := nil;
end;
If Cntrls <> nil then
begin
    A := 0;
    while ((LeftVol = -1) OR (Mute = -1)) AND (A < Cntrls.Count) do
    begin
        Cntrl := Cntrls[A];
        If Cntrl <> nil then
        begin
            If ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) OR
                ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND (Mute
= -1))) AND
                (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
            then
                begin
                    if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) then
                        MCD.cbStruct := SizeOf(TMixerControlDetails)
                    else
                        MCD.cbStruct := SizeOf(TMixerControlDetails);
                    MCD.dwControlID := Cntrl.dwControlID;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                        MCD.cChannels := 1
                    else
                        MCD.cChannels := ML.cChannels;
                    MCD.cMultipleItems := 0;
                    MCD.cbDetails := SizeOf(Integer);
                    MCD.paDetails := @details;
                    B := mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSf_VALUE);
                    If B = MMSYSERR_NOERROR then
                        begin
                            If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND
                                (LeftVol = -1) then
                                begin
                                    VolDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                                    LeftVol := details[0];
                                    If MCD.cChannels > 1 then
                                        begin
                                            RightVol := Details[1];
                                            Stereo := True;
                                        end
                                    else
                                        RightVol := LeftVol;
                                    end
                                else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE)
                                    AND (Mute = -1) then
                                        begin
                                            MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                                            If Details[0] <> 0 then
                                                Mute := 1
                                            else
                                                Mute := 0;
                                            end
                                        // NEW ->
                                        (*
                                            else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF)
                                        AND (Mute = -1) then
                                            begin
                                                MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
                                                If Details[0] <> 0 then
                                                    Mute := 1
                                                else
                                                    Mute := 0;
                                                MuteIsSelect := True;
                                            end;*)
                                end;*)
                end;
            end;
        end;
    end;
end;

```



```

MCD.cbDetails := 4;
MCD.paDetails := @Details;
MuteSet := True;
mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
  For B := 0 to Cntrl.cMultipleItems-1 do
    Details[B] := 0;

  GetMem (ltext, MCD.cChannels * MCD.cMultipleItems * sizeof
(MIXERCONTROLDETAILSF_LISTTEXT));
  MCDText.cbStruct := sizeof (MCDText);
  MCDText.dwControlID := Cntrl.dwControlID;
  MCDText.cChannels := MCD.cChannels;
  MCDText.cMultipleItems := MCD.cMultipleItems;
  MCDText.cbDetails := sizeof (MIXERCONTROLDETAILSF_LISTTEXT);
  MCDText.paDetails := ltext;
  mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
  B := MCD.cChannels - 1;
  while (B < integer (MCD.cMultipleItems)) do
  begin
    if (ltext[B].dwParam1 = MC.Data.dwLineID) then
      break;
    Inc (B, MCD.cChannels);
  end;
  FreeMem (ltext);

  If (B < integer (MCD.cMultipleItems)) then
  begin
    Details[B] := Mute;
    mixerSetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
    break;
  end;
end;
Inc (A);
end;
end;
end;
end;

If AConnection = -1 then
begin
  Cntrls := MD.Controls;
  ML := MD.Data;
end
else
begin
  If MC <> nil then
  begin
    Cntrls := MC.Controls;
    ML := MC.Data;
  end
  else
    Cntrls := nil;
end;
If Cntrls <> nil then
begin

  A := 0;
  while (not VolSet OR not MuteSet) AND (A < Cntrls.Count) do
  begin
    Cntrl := Cntrls[A];
    If Cntrl <> nil then
    begin

```

```

    If (((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND not
VolSet) OR
        ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND not
MuteSet) (* NEW -> *) (*OR
        ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) AND not
MuteSet)*) (* <- NEW *) AND
        (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
    then
    begin
        MCD.cbStruct := SizeOf(TMixerControlDetails);
        MCD.dwControlID := Cntrl.dwControlID;
        If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
            MCD.cChannels := 1
        else
            MCD.cChannels := ML.cChannels;
        MCD.cMultipleItems := 0;
        MCD.cbDetails := SizeOf(Integer);
        MCD.paDetails := @Details;
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) then
            begin
                Details[0] := LeftVol;
                If RightVol = -1 then
                    Details[1] := LeftVol
                else
                    Details[1] := RightVol;
                VolSet := True;
            end
        else if ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) (*
NEW -> *) (* OR
                (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) *)
        (* <- NEW *) then
            begin
                For B := 0 to MCD.cChannels - 1 do
                    Details[B] := Mute;
                    MuteSet := True;
                end;
                mixerSetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
            end;
        end;
        Inc (A);
    end;

    end;
end;

function TAudioMixer.GetMute(ADestination, AConnection: Integer; var Mute:
Boolean):Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
    mlMixerLine : TMixerLine;
    Cntrl:PMixerControl;
    Cntrls:TMixerControls;
    ML:TMixerLine;
    A,B:Integer;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;

```

```

MC := nil;
Mute := False;
MD := Destinations[ADestination];
if MD <> nil then
begin
  if AConnection = -1 then
    mlMixerLine := MD.Data
  else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
        mlMixerLine := MC.Data
      else
        Exit;
    end;
end;

GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
GetMem(pmcMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

with mlcMixerLineControlsMute do
begin
  cbStruct := SizeOf(TMIXERLINECONTROLS);
  dwLineID := mlMixerLine.dwLineID;
  dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
  cControls := 1;
  cbmxcctrl := SizeOf(TMIXERCONTROL);
  pamxcctrl := pmcMixerControlMute;
end;

if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
begin
  with mcdMixerDataMute do
  begin
    cbStruct := SizeOf(TMIXERCONTROLDETAILS);
    dwControlID := pmcMixerControlMute^.dwControlID;
    cChannels := 1;
    cMultipleItems := 0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
    paDetails := pmcMixerDataUnsignedMute;
  end;

  if mixerGetControlDetails(FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR then
  begin
    Mute := pmcMixerDataUnsignedMute^.fValue = 1;
    Result := True;
  end;
end
else
begin
  If (AConnection <> -1) then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Result = False) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
          (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
        begin
          mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
          mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
          If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
            mcdMixerDataMute.cChannels := 1
          else

```

```

        mcdMixerDataMute.cChannels := ML.cChannels;
        If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
            mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
        else
            mcdMixerDataMute.cMultipleItems := 0;
            mcdMixerDataMute.cbDetails := 4;
            mcdMixerDataMute.paDetails := @Details;
            mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE);

            GetMem (ltext, mcdMixerDataMute.cChannels *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
            MCDText.cbStruct := sizeof (MCDText);
            MCDText.dwControlID := Cntrl.dwControlID;
            MCDText.cChannels := mcdMixerDataMute.cChannels;
            MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
            MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
            MCDText.paDetails := ltext;
            mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
            B := mcdMixerDataMute.cChannels - 1;
            while (B < integer (mcdMixerDataMute.cMultipleItems)) do
            begin
                if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                    break;
                Inc (B, mcdMixerDataMute.cChannels);
            end;
            FreeMem (ltext);

            If (B < integer (mcdMixerDataMute.cMultipleItems)) then
            begin
                Result := True;
                Mute := Details[B] <> 0;
                break;
            end;
            end;
            Inc (A);
        end;
    end;
end;
end;
end;

FreeMem (pmcMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
    mlMixerLine : TMixerLine;
    Cntrl: PMixerControl;
    Cntrls: TMixerControls;
    ML: TMixerLine;
    A, B: Integer;
    details: array [0..100] of Integer;
    ltext: ^ListTextArray;
    MCDText: TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;

```

```

MC := nil;
MD := Destinations[ADestination];
if MD <> nil then
begin
  if AConnection = -1 then
    mlMixerLine := MD.Data
  else
  begin
    MC := MD.Connections[AConnection];
    if MC <> nil then
      mlMixerLine := MC.Data
    else
      Exit;
  end;
end;

GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
GetMem(pmcMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

with mlcMixerLineControlsMute do
begin
  cbStruct := SizeOf(TMIXERLINECONTROLS);
  dwLineID := mlMixerLine.dwLineID;
  dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
  cControls := 0;
  cbmxcctrl := SizeOf(TMIXERCONTROL);
  pamxcctrl := pmcMixerControlMute;
end;

if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
begin
  with mcdMixerDataMute do
  begin
    cbStruct := SizeOf(TMixerControlDetails);
    dwControlID := pmcMixerControlMute^.dwControlID;
    cChannels := 1;
    cMultipleItems := 0;
    cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
    paDetails := pmcMixerDataUnsignedMute;
  end;

  if Mute then
    pmcMixerDataUnsignedMute^.fValue := 1
  else
    pmcMixerDataUnsignedMute^.fValue := 0;

  if
(mixerSetControlDetails(FMixerHandle, @mcdMixerDataMute, MIXER_SETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
    Result := True;
  end
  else
  begin
    If (AConnection <> -1) then
    begin
      Cntrl := MD.Controls;
      ML := MD.Data;
      If (MC <> nil) AND (Cntrl <> nil) then
      begin
        A := 0;
        while (Result = False) AND (A < Cntrl.Count) do
        begin
          Cntrl := Cntrl[A];
          If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
          begin
            mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
            mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then

```

```

        mcdMixerDataMute.cChannels := 1
    else
        mcdMixerDataMute.cChannels := ML.cChannels;
    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
        mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
    else
        mcdMixerDataMute.cMultipleItems := 0;
        mcdMixerDataMute.cbDetails := 4;
        mcdMixerDataMute.paDetails := @Details;
        if (mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) <> MMSYSERR_NOERROR) then
            begin
                Inc (A);
                continue;
            end;
        if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            For B := 0 to Cntrl.cMultipleItems-1 do
                Details[B] := 0;
            If Mute then
                begin
                    GetMem (ltext, mcdMixerDataMute.cChannels *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT));
                    MCDText.cbStruct := sizeof (MCDText);
                    MCDText.dwControlID := Cntrl.dwControlID;
                    MCDText.cChannels := mcdMixerDataMute.cChannels;
                    MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
                    MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
                    MCDText.paDetails := ltext;
                    mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
                    B := mcdMixerDataMute.cChannels - 1;
                    while (B < integer (mcdMixerDataMute.cMultipleItems)) do
                        begin
                            if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                                break;
                            Inc (B, mcdMixerDataMute.cChannels);
                        end;
                        FreeMem (ltext);

                        If (B < integer (mcdMixerDataMute.cMultipleItems)) then
                            Details[B] := 1;
                        end;
                    if (mixerSetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR) then
                        begin
                            Result := True;
                            break;
                        end;
                    end;
                    Inc (A);
                end;
            end;
            end;
            end;
            end;

        FreeMem (pmcMixerDataUnsignedMute);
        FreeMem (pmcMixerControlMute);
    end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mcdMixerDataPeak : TMIXERCONTROLDETAILS;
    pmcMixerControlPeak : PMIXERCONTROL;
    { pmcMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED; }

```

```

mlMixerLine : TMixerLine;
A:Integer;
Cntrl:TMixerControls;
Details:Array [1..100] of Integer;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  LeftPeak := 0;
  RightPeak := 0;
  MD := Destinations[ADestination];
  if MD <> nil then
    begin
      if AConnection = -1 then
        begin
          mlMixerLine := MD.Data;
          Cntrl := MD.Controls;
        end
      else
        begin
          MC := MD.Connections[AConnection];
          if MC <> nil then
            begin
              mlMixerLine := MC.Data;
              Cntrl := MC.Controls;
            end
          else
            Exit;
          end;
        GetMem(pmcMixerControlPeak, SizeOf(TMIXERCONTROL));

        A := 0;
        while (A < Cntrl.Count) do
          begin
            If (Cntrl[A].dwControlType AND MIXERCONTROL_CT_CLASS_MASK) =
MIXERCONTROL_CT_CLASS_METER then
              break;
            Inc (A);
          end;
          If A = Cntrl.Count then
            begin
              FreeMem(pmcMixerControlPeak);
              Exit;
            end;

            with mcdMixerDataPeak do
              begin
                cbStruct := SizeOf(TMIXERCONTROLDETAILS);
                dwControlID := Cntrl[A].dwControlID;
                cChannels := mlMixerLine.cChannels;
                If (Cntrl[A].fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE) =
MIXERCONTROL_CONTROLF_MULTIPLE then
                  cMultipleItems:=Cntrl[A].cMultipleItems
                else
                  cMultipleItems:=0;
                cbDetails := SizeOf(TMIXERCONTROLDETAILSSIGNED);
                paDetails := @Details;
              end;
              if
(mixerGetControlDetails(FMixerHandle,@mcdMixerDataPeak,MIXER_GETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
                begin
                  LeftPeak := Details[1];
                  if mlMixerLine.cChannels = 2 then
                    RightPeak := Details[2]
                  else
                    RightPeak := LeftPeak;
                  Result := True;
                end;
            end;

```

```
    FreeMem(pmcMixerControlPeak);  
end;  
end;  
procedure Register;  
begin  
    RegisterComponents('Samples', [TAudioMixer]);  
end;  
end.
```

Кафедра КБПЗ — 2021 рік